TESE DE MESTRADO

TÉCNICAS DE DECISÃO SUAVE

Hélio Magalhães de Oliveira

COORDENAÇÃO DO MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CIDADE UNIVERSITÁRIA
RECIFE - BRASIL

- 1983 -

TESE DE MESTRADO

TÉCNICAS DE DECISÃO SUAVE

Hélio Magalhães de Oliveira

COORDENAÇÃO DO MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CIDADE UNIVERSITÁRIA
RECIFE - BRASIL

- 1983 -

TESE DE MESTRADO

TÉCNICAS DE DECISÃO SUAVE

Hélio Magalhães de Oliveira

COORDENAÇÃO DO MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CIDADE UNIVERSITÁRIA
RECIFE - BRASIL

- 1983 -

UNIVERSIDADE FEDERAL DE PERNAMBUCO
COORDENAÇÃO DO MESTRADO EM ENGENHARIA ELÉTRICA

TÉCNICAS DE DECISÃO SUAVE

Hélio Magalhães de Oliveira

Tese apresentada a Coordenação do Mestrado em Engenharia Elétrica da Universidade Federal de Pernambuco em cumprimento a exigências para obtenção do título de Mestre em Ciências.

Orientador: *Valdemar Cardoso da Rocha Junior*

Junho - 1983

Aos meus Pais,

Nilson e Djanira.

AGRADECIMENTOS

Gostaria de expressar meus sinceros agradecimentos a todos que prestaram sua colaboração na elaboração deste trabalho.

A todos os professores do curso de Mestrado em Engenharia Elétrica, em especial aos professores Marco Wolfenson, Fernando Campello e Luiz Torres.

Aos colegas Marcos Antonio Martins, Clóvis P. Florêncio (entre outros), pela suas contribuições.

A Ricardo Campello e Fernando Casanova pelo auxílio na parte computacional e sugestões valiosas.

A senhora Rosa Maria Alves Neves, pela cuidadosa datilografia dos originais manuscritos.

Ao professor Valdemar Cardoso da Rocha Junior, por introduzir-me neste campo de pesquisa, por sua influência na minha filosofia, sua orientação, incentivo e amizade.

RESUMO

Nesta tese são apresentados e analisados alguns recentes resultados obtidos na área de códigos corretores de erros, relacionados com decodificação utilizando decisão suave. Tal procedimento é usado com o fim de evitar a degradação no desempenho de sistemas codificados que resulta quando na recepção uma quantização abrupta precede a decodificação.

No capítulo I é apresentado um tratamento introdutório sobre códigos lineares, estudando códigos de bloco e códigos convolucionais. Em seguida, no capítulo II, é abordado o conceito de decisão suave e é apresentada uma maneira de utilizá-la, adaptando técnicas algébricas de decodificação por distância mínima a uma medida de distância que permite o uso de informação de verossimilhança. Alguns procedimentos subótimos que utilizam decisão suave são também apresentados. No capítulo III são estudados dois algorítmos ótimos que fazem uso de tais técnicas na decodificação de códigos lineares em canais sem memória. O desempenho destes algorítmos é analisado através de simulação em computador digital, no capítulo IV, para vários códigos. Os resultados obtidos são comparados com procedimentos que empregam decisão abrupta. Finalmente, no capítulo V, comentários gerais são feitos discutindo vantagens e restrições decorrentes do uso destas técnicas, assim como sugestões para investigações futuras.

ABSTRACT

In this thesis, some recent results obtained in the area of error correcting codes, related to soft decision decoding techniques, are presented and analyzed. Such procedure is used in order to avoid the degradation in the performance of encoded systems which results when in the reception a hard decision proceedes the decoding.

In chapter I an introductory treatment of linear codes is presented, studying block codes and convolutional codes. Then, in, chapter II, the concept of soft decision is approached, and a way of using it is presented, adapting algebraic techniques of minimum distance decoding to a distance measure which permits the use of likelihood information. Some suboptimal procedures which use soft decision are also presented.

In chapter III two algorithms with optimal properties are studied, which make use of such techniques in the decoding of linear codes in memoryless channels. The performance of these algorithms is analyzed via digital computer simulation, in chapter IV, for several codes. The results obtained are compared with procedures which use hard decision. Finally, in chapter V, general comments are made discussing the advantages and restrictions implied by the use of these techniques, as well as suggestions for further investigations.

Í N D I C E

CAPÍTULO I

INTRODUÇÃO AOS CÓDIGOS LINEARES..... 1

1.1 - INTRODUÇÃO..... 1

1.2 - CÓDIGOS DE BLOCO..... 4

1.2.1 - Códigos de Bloco Lineares..... 4

1.2.2 - Capacidade de Correção..... 5

1.2.3 - Matriz Geradora de um Código Linear..... 7

1.2.4 - A Matriz de Verificação de Paridade..... 10

1.2.5 - Síndrome..... 12

1.2.6 - Códigos Cíclicos Binários..... 13

1.2.7 - Códigos de Hamming e Códigos BCH..... 16

1.3 - CÓDIGOS CONVOLUCIONAIS..... 17

1.3.1 - Considerações Gerais..... 17

1.3.2 - O Codificador Convolucional..... 18

1.3.3 - Outra Representação da Matriz Geradora..... 22

1.3.4 - Diagrama de Árvore, Treliça e Diagrama de Estado.. 25

1.3.5 - Decodificação de Códigos Convolucionais..... 27

CAPÍTULO II

DECODIFICAÇÃO USANDO DECISÃO SUAVE..... 29

2.1 - INTRODUÇÃO..... 29

2.2 - ALGUNS PROCEDIMENTOS SUBÓTIMOS..... 35

2.2.1 - Decisão Suave na Decodificação de Códigos com um Único Dígito de Paridade..... 35

2.2.2 - Decodificação por Distância Suave Mínima..... 37

2.2.3 - Algorítmo de Harrison..... 38

2.3 - DECODIFICAÇÃO POR DECISÃO SUAVE PARA CÓDIGOS DE BLOCO USANDO UMA TRELIÇA..... 41

2.3.1 - Considerações Gerais..... 43

2.3.2 - Treliça Associada a um Código de Bloco Linear..... 43

2.3.3 - Construção da Treliça..... 45

2.3.4 - Decodificador de Viterbi usando a Treliça..... 48

2.4 - DECISÃO SUAVE NA DECODIFICAÇÃO QUANDO A FONTE TEM DISTRI - BUIÇÃO DE PROBABILIDADE DESCONHECIDA..... 49

2.4.1 - Noções Preliminares..... 49

2.4.2 - Decodificação usando Confiança Condicional..... 50

2.4.3 - Exemplo: Aplicação a um Canal com Ruído Branco Gaussiano..... 52

2.5 - DISTÂNCIA GENERALIZADA..... 54

2.5.1 - Medidas de Distância..... 54

2.5.2 - Decodificação por Distância Mínima Generalizada I. 56

2.5.3 - Decodificação por Distância Mínima Generalizada II 64

2.6 - DECODIFICAÇÃO POR DECISÃO SUAVE EMPREGANDO SUBTRELIÇAS.... 74

CAPÍTULO III

DECODIFICAÇÃO ÓTIMA DE CÓDIGOS LINEARES..... 77

3.1 - O ALGORÍTMO DE HARTMANN E RUDOLPH..... 78

3.1.1 - Introdução..... 78

3.1.2 - A Regra de Decodificação..... 79

3.1.3 - Exemplos..... 83

3.1.4 - Desempenho Assintótico do Algorítmo..... 87

3.2 - ALGORÍTMO PARA MAXIMIZAÇÃO DA PROBABILIDADE A POSTERIORI.. 91

3.2.1 - Introdução..... 91

3.2.2 - A Regra de Decodificação..... 93

3.2.3 - Aplicações para Códigos Lineares..... 96

3.2.4 - Análise do Comportamento Assintótico..... 100

CAPÍTULO IV

SIMULAÇÃO EM COMPUTADOR..... 104

4.1 - ANÁLISE DO DESEMPENHO DO ALGORÍTMO DE HARTMANN-RUDOLPH... 105

4.1.1 - Considerações Gerais..... 105

4.1.2 - Curvas de Desempenho..... 107

4.2 - DESEMPENHO DA DECODIFICAÇÃO UTILIZANDO MÁXIMA PROBABILIDADE A POSTERIORI..... 125

4.2.1 - Considerações Gerais..... 125

4.2.2 - Implementação Computacional da Treliça..... 126

4.2.3 - Curvas de Desempenho..... 130

CAPÍTULO V

CONCLUSÕES..... 137

5.1 - ANÁLISE DOS RESULTADOS..... 137

5.2 - COMENTÁRIOS..... 141

APENDICE A..... 146

APENDICE B..... 156

APENDICE C..... 159

REFERÊNCIAS E BIBLIOGRAFIA..... 162

ÍNDICE DE FIGURAS

Figura 1.1 - CODIFICADOR PARA O CÓDIGO CONVOLUCIONAL(2,1,3).... 24

Figura 1.2 - CODIFICADOR PARA O CÓDIGO CONVOLUCIONAL(3,2,2).... 24

Figura 1.3 - DIAGRAMA EM ÁRVORE PARA O CÓDIGO CONVOLUCIONAL (4,1,3)..... 25

Figura 1.4 - DIAGRAMA DE TRELIÇA PARA O CÓDIGO CONVOLUCIONAL (4,1,3)..... 26

Figura 1.5 - DIAGRAMA DE ESTADO PARA O CÓDIGO CONVOLUCIONAL (4,1,3)..... 27

Figura 2.1 - DECISÃO SUAVE-QUANTIZAÇÃO EM 4 REGIÕES DE CONFIABILIDADE..... 31

Figura 2.2 - DETEÇÃO POR ZONA NULA..... 31

Figura 2.3 - SAÍDAS DO DEMODULADOR EMPREGANDO DECISÃO SUAVE.... 33

Figura 2.4 - CURVAS DE DESEMPENHO DO ALGORÍTMO DE FARRELL-KALLIGEROS..... 36

Figura 2.5 - CÓDIGO PRODUTO..... 38

Figura 2.6 - CURVAS DE DESEMPENHO NA DECODIFICAÇÃO POR DISTÂNCIA SUAVE MÍNIMA..... 39

Figura 2.7 - REGIÕES DE CONFIABILIDADE..... 40

Figura 2.8 - CURVAS DE DESEMPENHO PARA O ALGORÍTMO DE HARRISON. 42

Figura 2.9 - DIAGRAMA DE ÁRVORE PARA UM CÓDIGO DE BLOCO..... 44

Figura 2.10 - TRELIÇA NÃO EXPURGADA PARA O CÓDIGO (5,3,2)..... 47

Figura 2.11 - TRELIÇA PARA O CÓDIGO DE BLOCO (5,3,2)..... 47

Figura 2.12 - PASSOS NA DECODIFICAÇÃO DO CÓDIGO (5,3,2) USANDO O MÉTODO DE WOLFENSON-ROCHA..... 53

Figura 2.13 - PARTIÇÃO NÃO SIMÉTRICA..... 70

Figura 2.14 - PASSOS NA DECODIFICAÇÃO POR DISTÂNCIA MÍNIMA GENERALIZADA..... 74

Figura 2.15 - EXEMPLO DE SUBTRELIÇAS PARA UM CÓDIGO DE BLOCO.... 75

Figura 3.1 - ESPAÇO LINHA DO CÓDIGO DUAL DO CÓDIGO DE HAMMING (7,4,3) 84

Figura 3.2 - DECODIFICADOR ÓTIMO PARA O CÓDIGO DE BLOCO(7,4,3) .. 84

Figura 3.3 - ESPAÇO LINHA DO CÓDIGO DUAL DO CÓDIGO CONVOLUCIONAL (4,3,3) 85

Figura 3.4 - DECODIFICADOR ÓTIMO PARA O CÓDIGO CONVOLUCIONAL (4,3,3) 86

Figura 3.5 - DENSIDADES DE PROBABILIDADE DA ENVOLTÓRIA..... 97

Figura 3.6 - TRELIÇA ASSOCIADA AO CÓDIGO LINEAR(3,2,2)..... 98

Figura 3.7 - PASSOS NA DECODIFICAÇÃO PARA O CÓDIGO (3,2,2)..... 99

Figuras 4.1 a 4.17 - CURVAS DE DESEMPENHO PARA O ALGORÍTMO DE HARTMANN-RUDOLPH.....108-124

Figuras 4.18 a 4.23- CURVAS DE DESEMPENHO PARA O ALGORÍTMO DE MAXIMIZAÇÃO DA PROBABILIDADE A POSTERIORI.....131-136

Figuras 5.1 a 5.2 - DESEMPENHO DA DECODIFICAÇÃO POR DISTÂNCIA GENERALIZADA.....139-140

Figuras 5.3 a 5.4 - VERIFICAÇÃO DA SIMETRIA DO RUÍDO GAUSSIANO GERADO.....142-143

LISTA DE SÍMBOLOS E ABREVIATURAS

A	Amplitude de um pulso binário
$AB_{\gamma b \to 0}$	Comportamento assintótico para baixa relação sinal-ruído
$AB_{\gamma b \to \infty}$	Comportamento assintótico para alta relação sinal-ruído
ASK	Modulação por chaveamento de amplitude
BCH	Bose-Chaudhuri-Hocquenghem
BEC	Canal binário com apagamento
BSC	Canal binário simétrico
$\underline{c}$	Palavra código, ou sequência código
$c_n^i = n!/i!(n-i)!$	
C	Código de bloco
C'	Código dual do código linear C
d	Distância mínima de um código de bloco
d_s	Distância suave
$D_G(\underline{r},\underline{f})$	Distância generalizada de Forney
$D_H(\underline{f},\underline{g})$	Distância de Hamming entre duas énuplas
$\underline{e}_m = (\delta_{0m}, \delta_{1m}, \ldots, \delta_{n-1m})$	
$\underline{f}$	Palavra código de um código de bloco
fdP	Função densidade de probabilidade
$f(r\|c)$	Função de verossimilhança de um canal
$\underline{g}$	Palavra código de um código de bloco
g(i)	Geradores de um código convolucional
g(i,j)	Subgeradores de um código convolucional
g(s)	Função geradora de momentos
g(x)	Polinômio gerador de um código cíclico
$[G], G_\infty$	Matriz geratriz de um código linear

GF(q)	Campo de Galois com q elementos
h(x)	Polinômio de verificação de paridade de um código cíclico
$[H], H_\infty$	Matriz de verificação de paridade de um código linear
HF	Alta frequência
iid	Independentes e identicamente distribuídas
I_k	Matriz identidade kxk
$I_0(\ )$	Função de Bessel modificada de ordem zero(primeira espécie)
J	Amplitude da zona nula; número total de classes de confiabilidade
k	Números de dígitos de informação; profundidade em uma treliça
L_j	log razão de verossimilhança na classe de confiabilidade c_j
(m)	Bloco de mensagem
m(X)	Polinômio mensagem
n	Comprimento de um código de bloco
ndc	Não decodificar corretamente
N	Potência do ruído
N(d)	Número de palavras código com peso d
NP	Neymann-Pearson
$p_0, p_1, \ldots$	Probabilidades a priori dos símbolos da fonte
P( ), Pr( )	Probabilidade de um evento
$P\{decj \mid \Pi^s, j\}$	Probabilidade de decodificar j dado que Π^s foi observado e j foi transmitido
P_k	Subconjunto de $\{0,1,\ldots,q^{n-k}-1\}$
PLL	Phase lock loop
2^Q	Número de níveis de quantização
$Q(x) = 1/\sqrt{2\Pi}\int_x^\infty \exp(-t^2/2)\,dt$	
$\underline{r}$	Palavra recebida
rem	Resto de uma divisão

$r(X)$	Polinômio recebido
R	Taxa (eficiência) de um código corretor de erros
R_{Si}	Coeficiente de confiabilidade condicional de Kiefer
$\mathbb{R}$	Conjunto dos números reais
R_Q	Espaço das possíveis regiões de quantização
$R_{\emptyset}$	Espaço dos possíveis valores do ℓog verossimilhança
$\underline{S}$	Síndrome
$\underline{S}_{\ell}(k)$	ℓ-ésimo nó na profundidade k
$S;s$	Conjunto de índices; número de apagamentos
SNR	Relação sinal ruído
t	Capacidade de correção de um código
$\underline{u},\underline{v}$	Énuplas sobre GF(q)
V_n	Espaço vetorial das énuplas sobre GF(q)
$W(\underline{u})$	Peso da énupla $\underline{u}$
$\hat{X}$	Estimador de X
$\underline{X}$	Énupla
$(X),[X]$	Matriz X
$(X)^T,[X]^T$	Transposta de $[X]$
$\lfloor X \rfloor$	Parte inteira de X
$Z(\underline{c})$	ℓog da função de verossimilhança $f(\underline{r}\|\underline{c})$
α	Probabilidade de erro tipo I
α_j,α_s	Peso de uma classe de confiabilidade
β	Probabilidade de erro tipo II
γ,γ_b	Relação sinal-ruído de um canal ruidoso
δ_{ij}	Delta de Kronecker
$\Theta = \exp(2\Pi\sqrt{-1}/P)$	
λ	Constante
$\mu(s)$	Função geradora cumulativa
Π	Partição do espaço de observações
$\rho = (1-\emptyset)/(1+\emptyset)$	
σ	Tensão eficaz do ruído

$\emptyset_j$	Razão de verossimilhança para a amostra r_j
$\emptyset_{\ell \mid j}$	Razão de verossimilhança para a classe Π_{ℓ}^{s} dado j
Σ	Somatório módulo-q
$\oplus$	Soma módulo-2
$\mathcal{N}$	Distribuição normal
$\mathcal{B}$	Distribuição binomial

CAPÍTULO I

INTRODUÇÃO AOS CÓDIGOS LINEARES

1.1 - INTRODUÇÃO

Sistemas de comunicação são essencialmente usados para transmissão de informação, e em geral são constituídos por um transmissor, um meio de transmissão através do qual a informação é enviada e um receptor. Ao bloco transmissor está conectada a fonte de informação a ser transmitida - sinais de voz, sinais de TV, dados de saída de computadores, dados de telemetria de naves de exploração espacial ou dados de telemetria transmitidos de uma instalação operada automaticamente para uma estação central de controle.

À medida que os sinais atravessam o meio de transmissão (canal), eles são distorcidos; ruídos e interferências a eles se somam e se torna uma tarefa importante interpretar os sinais quando finalmente recebidos no extremo receptor.

O problema básico em um sistema de comunicação se resume em transmitir a informação através do canal ruidoso, de uma maneira confiável.

O problema da transmissão confiável de informação tem representado um desafio constante para engenheiros e pesquisadores em comunicações. Aqui, a confiabilidade diz respeito à imunidade da transmissão a ação do ruído e outras interferências, e não a indeci

frabilidade de mensagens (criptografia).

Nas últimas décadas tem havido um grande crescimento na utilização de sistemas de comunicação digital, i.e., sistemas onde a mensagem já está na forma digital ou é convertida para este formato, como no caso dos sistemas PCM/TDM |33|. Um sistema de comunicação digital pode ser descrito como sendo constituído pelos seguintes elementos: Fonte → Codificador → Canal → Decodificador → Destinatário.

A fonte consiste da fonte original de informação e de um codificador fonte, enquanto que o canal consiste de um modulador, um meio de transmissão e um demodulador. As razões para o crescimento do emprego de sistemas de comunicação digital são bastante conhecidas |11|. Para estes sistemas, uma abordagem eficaz para o problema da confiabilidade envolve o uso de códigos corretores de erros. Será mostrado que a introdução de símbolos adicionais (redundantes) ao fluxo de símbolos produzidos pela fonte (dígitos de informação) permitirá que se execute a deteção e correção de erros às custas, evidentemente, de um aumento na complexidade do sistema e de uma redução na taxa de transmissão de dados (ou de um aumento na banda passante exigida). A potencialidade máxima dos códigos corretores de erros foi estabelecida em 1948 por C.E.Shannon |35| através do teorema de codificação para um canal ruidoso. Este teorema estabelece que todo canal tem uma capacidade máxima C e para qualquer taxa de transmissão $R < C$, existem códigos de taxa R, os quais, com decodificação de máxima verossimilhança, tem uma probabilidade de decodificação errônea arbitrariamente pequena. O teorema de Shannon demonstrou a existência de códigos os quais resultam numa probabilidade de erro na decodificação arbitrariamente pequena, mas não indicou como construir tais códigos.

Devido a presença de sinais ruidosos no canal, como ante

riormente mencionado, podem ocorrer erros durante a transmissão. Estes erros podem ser esporádicos e independentes, neste caso são chamados erros aleatórios, ou podem ocorrer em "burst" de vários erros de cada vez, e então diz-se que o canal tem memória. O último caso não será objeto de estudo neste trabalho.

Para combater os possíveis erros que possam ocorrer durante uma transmissão de dados digitais em um canal de comunicação ruidoso, são empregados códigos corretores de erros, os quais podem ser classificados como códigos lineares ou não lineares. Códigos lineares têm os seus dígitos redundantes calculados como soma módulo q |36| dos dígitos de informação, enquanto que códigos não lineares empregam lógica não linear (tais como portas e, ou, não, etc , no caso binário). No que segue serão abordados os códigos lineares, por sua importância prática e por serem estudados na quase totalidade das publicações sobre códigos.

Entre os códigos lineares existem basicamente duas técnicas distintas de controle de erros para realização de uma transmissão confiável de dados digitais em canais ruidosos: Os códigos de bloco e os códigos de árvore. Dependendo da maneira como a redundância é adicionada aos dígitos de informação (mensagem), um destes dois tipos de código é gerado. Códigos para os quais a redundância em um bloco de dígitos verifica a ocorrência ou não de erros, apenas naquele bloco particular, são chamados códigos de bloco. Já os códigos onde a redundância em um bloco verifica a existência ou não de erros em mais de um bloco são chamados de códigos de árvore. A subclasse mais importante destes códigos é a dos códigos convolucionais, os quais são mais simples e de fácil implementação com relação a outros tipos de códigos de árvore.

Os códigos de bloco e convolucionais são competitivos em muitas situações. A escolha final de um deles depende de fatores como o formato dos dados, o retardo na decodificação, a complexida-

de do sistema necessário para alcançar uma determinada taxa de erros, etc. É dada ênfase neste trabalho aos códigos de bloco, muito embora os algorítmos considerados sejam na sua maioria também aplicáveis a códigos convolucionais.

Nos anos recentes tem havido um grande interesse em esquemas de decodificação utilizando decisão suave, aplicáveis a estes dois tipos de códigos, com a finalidade de evitar a degradação no desempenho que resulta quando uma quantização abrupta símbolo-por-símbolo precede o decodificador. Em sistemas de comunicação via satélite, por exemplo, uma diminuição de 1 dB na relação sinal/ruído proprociona uma economia de milhões de dólares.

O intuito desta tese é de apresentar, sugerir e analisar alguns importantes algorítmos de decodificação de códigos lineares que façam uso de tais técnicas. Neste capítulo são estudados de forma introdutória os códigos lineares, de modo que seja possível a compreensão das técnicas de decodificação probabilísticas abordadas em capítulos posteriores.

1.2 - CÓDIGOS DE BLOCO

1.2.1 - Códigos de Bloco Lineares

Os códigos de bloco lineares representam sem dúvida a parte mais bem desenvolvida da teoria dos códigos corretores de erros, fato que se deve a sua forte estrutura algébrica, que permite o emprego de ferramentas matemáticas como a teoria das matrizes, e a teoria dos campos de Galois, por exemplo.

O processo de codificação consiste em segmentar uma mensagem em blocos de k dígitos e acrescentar a cada bloco $n-k$ dígitos redundantes. Estes $n-k$ dígitos são determinados a partir dos k dígitos de mensagem e destinam-se a deteção e/ou correção de erros que possam vir a ocorrer durante a transmissão. Des-

te modo, o codificador opera de modo independente com cada bloco , e a redundância acrescentada serve apenas para detetar e/ou corrigir erros somente no bloco considerado.

Um fato importante a ser considerado no estudo de códigos de bloco, é que para um comprimento de bloco n e uma taxa k/n fixadas, o melhor código de bloco linear tem desempenho quase tão bom quanto o melhor código de bloco com os mesmos parâmetros |5|. A utilização de códigos lineares resulta em simplificação na implementação e na análise do desempenho, por isso são os estudados quase na totalidade das pesquisas na área.

Alguns códigos de bloco simples e bastante utilizados, tais como códigos de repetição, códigos de peso constante, códigos de arranjos, códigos de único dígito de paridade e códigos de Hamming estão descritos nas referências |5|,|11|.

Definição 1.2.1 - Um código de bloco linear $C(n,k,d)$ é um conjunto de q^k énuplas com elementos em $GF(q)$, chamadas de palavras código, as quais diferem entre si em pelo menos d posições e formam um subespaço do espaço vetorial V_n de todas as énuplas definido sobre $GF(q)$.

Deve ser notado conforme definido acima, que o codificador se torna proibitivamente complexo para valores grandes de k, desde que devem ser armazenadas as q^k palavras código de n dígitos cada, em uma memória. Isto pode ser evitado lembrando que o código é um subespaço vetorial, como será visto a seguir.

1.2.2 - Capacidade de Correção

Dado um código de bloco $C(n,k,d)$, é importante determinar qual a sua capacidade de correção, i.e., quantos erros ele é capaz de detetar/corrigir, em cada bloco transmitido. Será mostrado que esta capacidade de correção está relacionada com o parâmetro d.

Portanto, uma importante figura de mérito de um código é a sua distância mínima d entre duas palavras código, medida em termos de distância de Hamming abaixo definida.

Definição 1.2.2a - O peso de Hamming de uma énupla $\underline{u}$, denotado por $W(\underline{u})$, é definido como sendo o número de componentes não nulas de $\underline{u}$.

Assim, por exemplo, se $\underline{u} = (1,0,2,0,0,5,0,1)$, então $W(\underline{u}) = 4$.

Definição 1.2.2b - A distância de Hamming entre duas énuplas $\underline{u}$ e $\underline{v}$ denotada por $D_H(\underline{u},\underline{v})$, é definida como o número de componentes nas quais estas énuplas diferem.

Assim, se $\underline{u} = (1,0,2,0,1,2)$ e $\underline{v} = (1,0,1,0,2,2)$, a distância de Hamming entre elas é $D_H(\underline{u},\underline{v}) = 2$.

Se os vetores considerados são binários, da definição de soma módulo 2 é fácil verificar que $D_H(\underline{u},\underline{v}) = W(\underline{u} \oplus \underline{v})$, i.e., que a distância entre $\underline{u}$ e $\underline{v}$ é exatamente igual ao peso do vetor soma.

Dado um código linear $C(n,k,d)$, calculando as distâncias entre todos os possíveis pares de palavras código, a menor distância obtida é chamada de distância mínima do código, denotada por d. Portanto,

$$d = \min_{\underline{u} \neq \underline{v}} D_H(\underline{u},\underline{v}) \qquad \underline{u},\underline{v} \in C \tag{1-1}$$

Se $\underline{u}$ e $\underline{v}$ são palavras código de um código linear, então $\underline{u} \oplus \underline{v}$ também o é, visto que o código é um subespaço vetorial. Segue-se deste fato que a distância mínima para um código linear binário é igual ao peso mínimo dentre as palavras código não nulas.

Agora, com relação à capacidade de deteção de um código de bloco $C(n,k,d)$, torna-se trivial verificar que a ocorrência

de uma quantidade de erros menor ou igual a $d-1$ resulta em uma das 2^n-2^k énuplas que não são palavras código, o que possibilita a deteção de erros. Portanto, um código com distância mínima d é capaz de detetar até $(d-1)$ erros.

Pode ser facilmente mostrado |36| que na decodificação por máximo de verossimilhança com canal BSC, o decodificador estima a palavra transmitida como sendo a palavra código mais próxima da palavra recebida no sentido de distância de Hamming. Se o código de bloco é usado para correção de erros aleatórios e tem distância mínima tal que $2t+2 \geq d \geq 2t+1$, então este decodificador poderá corrigir todas as configurações de erros contendo t ou menos erros que possam ter ocorrido |36|.

Assim, um código com distância mínima d tem uma capacidade corretora de $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ erros.

Desta forma o problema de encontrar um código com uma dada capacidade de correção, consiste basicamente em garantir uma dada distância mínima. A teoria da informação proporciona o estabelecimento de cotas superiores e inferiores para d, quando são fixados um comprimento n e uma taxa k/n, contudo não indica como construir tais códigos |15|.

1.2.3 - Matriz Geradora de um Código Linear

Para um subespaço C de V_n constituindo um código linear, é possível encontrar um conjunto de k énuplas linearmente independentes, as quais constituem uma base para C. Então, cada énupla $\underline{u} \in C$ pode ser expressa como combinação linear dos vetores $\{\underline{V}_1, \underline{V}_2, \ldots, \underline{V}_k\}$ da base, i.e.,

$$\underline{u} \in C \Rightarrow \underline{u} = m_1\underline{V}_1 \oplus_q m_2\underline{V}_2 \oplus_q \cdots \oplus_q m_k\underline{V}_k \qquad (1\text{-}2)$$

ou seja,

$$\underline{u} \in C \Rightarrow \underline{u} = \sum_{\ell=1}^{k} m_\ell \underline{V}_\ell, \ (\forall \ell)\ m_\ell \in GF(q) \qquad (1\text{-}3)$$

Um código linear $C(n,k,d)$ constitui um subespaço de V_n com dimensão k, de modo que é possível representá-lo através dos k vetores que constituem a base. Estes vetores $\{\underline{V}_\ell\}_1^k$ podem ser arranjados como linhas de uma matriz $k \times n$, denominada de matriz geratriz, como mostrado abaixo.

$$[G] = \begin{vmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \vdots \\ \underline{V}_k \end{vmatrix} = \begin{vmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & & \vdots \\ V_{k1} & V_{k2} & \cdots & V_{kn} \end{vmatrix} \tag{1-4}$$

Se é assumido um bloco de informação $\underline{m} = (m_1,\ldots,m_k)$, então a palavra código correspondente é dada por $\underline{u} = \underline{m}[G]$;

$$\underline{u} = (m_1,\ldots,m_k) \begin{vmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \vdots \\ \underline{V}_k \end{vmatrix} = \sum_{\ell=1}^{k} m_\ell \, \underline{V}_\ell \tag{1-5}$$

ou seja, a palavra código correspondente à mensagem $\underline{m}$ é uma combinação linear de linhas de $[G]$.

Assim, as linhas da matriz geradora $[G]$ geram um código linear $C(n,k,d)$, onde cada bloco de informação é codificado em uma palavra com n dígitos para transmissão. A razão $R = k/n$ é chamada de taxa (ou eficiência) do código.

Desde que um código linear é completamente especificado pela matriz $[G]$, o tamanho da memória requerida no codificador é reduzido enormemente. O codificador pode armazenar somente as k linhas de $[G]$, ao invés das q^k palavras código, se ele possuir elemento lógico capaz de fazer as combinações lineares das linhas

armazenadas.

Exemplo 1.1 - Seja um código de bloco linear (5,3,2) com matriz geratriz dada por

$$[G] = \begin{vmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \underline{V}_3 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{vmatrix}$$

A palavra código correspondente à mensagem $\underline{m} = (1,0,1)$, por exemplo, é encontrada facilmente por

$$\underline{u} = (1,0,1) \begin{vmatrix} \underline{V}_1 \\ \underline{V}_2 \\ \underline{V}_3 \end{vmatrix} = 1\underline{V}_1 \oplus 0\underline{V}_2 \oplus 1\underline{V}_3 = (1,0,1,1,0).$$

É possível codificar cada bloco de informação em palavras código de tal maneira que os k primeiros dígitos da palavra código sejam exatamente os mesmos do bloco de informação, e os últimos $n-k$ dígitos sejam os dígitos redundantes que são função dos dígitos de informação. Um código nesta forma é dito sistemático. A redundância é adicionada de maneira a proporcionar capacidade de proteção à mensagem, e o problema do codificador se reduz a determinar estes dígitos.

Um código sistemático pode ser descrito por uma matriz geradora [G] na forma

$$[G] = \begin{vmatrix} 1 & 0 & \dots & 0 & P_{1,1} & \dots & P_{1,n-k} \\ 0 & 1 & \dots & 0 & P_{2,1} & \dots & P_{2,n-k} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & & 1 & P_{k,1} & \dots & P_{k,n-k} \end{vmatrix} \qquad (1\text{-}6)$$

Onde $p_{ij} \varepsilon GF(q)$.

Isto pode ser representado na forma $[G] = [I_k \vdots P]$, onde I_k é uma matriz identidade kxk e [P] uma matriz kx(n-k) com elementos em GF(q).

Se é considerado um bloco de informação $\underline{m} = (m_1, \ldots, m_k)$, e uma matriz geradora na forma acima, a palavra código correspondente será dada por

$$\underline{u} = (u_1, \ldots, u_n) = (m_1, \ldots, m_k) \cdot [G] \qquad (1\text{-}7)$$

Efetuando a multiplicação, são obtidas as equações de verificação de paridade do código:

$$\begin{aligned} u_j &= m_j & j=1,2,\ldots,k \\ u_{k+j} &= p_{1j}m_1 \oplus \ldots \oplus p_{kj}m_k & j=1,2,\ldots,n-k \end{aligned} \qquad (1\text{-}8)$$

Os k primeiros dígitos da palavra código são exatamente os dígitos de informação a serem transmitidos, enquanto que os últimos n-k dígitos são redundantes (ou de verificação de paridade), calculados como funções lineares dos dígitos de informação.

1.2.4 - A Matriz de Verificação de Paridade

Para cada matriz [G] kxn, existe uma matriz (n-k)xk [H] de tal forma que o espaço linha de [G] é ortogonal a [H], i.e., o produto interno de vetores do espaço linha de [G] com as linhas de [H] é nulo. Esta matriz é denominada de matriz de verificação de paridade e pode ser escrita sob a forma

$$[H] = [\underline{h}_1 \ \underline{h}_2 \ldots \underline{h}_{n-k}]^T, \text{ onde } (\forall i) \ \underline{h}_i \text{ é uma énupla} \qquad (1\text{-}9)$$

Desta forma, se [H] é o espaço nulo de [G], então se verifica a relação

$$[G].[H]^T = \underline{0} \qquad (1\text{-}10)$$

Se o código está na forma sistemática $[G] = [I_k \vdots P]$, então o espaço linha de [G] é o espaço nulo da matriz [H] dada por $[H] = [-P^T \vdots I_{n-k}]$. (1-11)

<u>Definição 1.2.4</u> - Se [H] é usada como matriz geratriz de um código de bloco, este código é dito ser o dual do código gerado pela matriz [G].

Seja $\underline{u} = (u_1,\ldots,u_n)$ um vetor qualquer do espaço linha de [G]. Então $\underline{u}\,[H]^T = \underline{0}$, pois desde que [H] é o espaço nulo de [G], $\underline{u}.\underline{h}_i = 0$ para $i=1,2,\ldots,n-k$.

Então o código linear gerado por [G] pode também ser descrito de outra maneira:

"$\underline{u}$ é palavra código se e somente se $\underline{u}[H]^T = \underline{0}$, onde [H] é a matriz de verificação de paridade do código linear gerado por [G]".

<u>Exemplo 1.2</u> - A matriz [H] para o código de bloco (5,3,2) introduzido no exemplo 1.1 pode ser determinada pela equação (1-11) resultando em

$$[H] = \left|\begin{matrix} 1 & 1 & 0 & \vdots & 1 & 0 \\ & & & \vdots & & \\ 1 & 0 & 1 & \vdots & 0 & 1 \end{matrix}\right|$$

A énupla $\underline{u} = (1,0,1,1,0)$ resulta em $\underline{u}[H]^T = 0$.

Conclui-se, portanto, que um código linear é unicamente especificado pela matriz geradora ou pela matriz de verificação de paridade.

Na construção de um código linear, a matriz [P] deve ser escolhida de modo que o código tenha as propriedades de correção de erros desejadas.

Para um código com matriz geradora $[G] = [I_k \vdots P]$, as equações de verificação de paridade podem também ser obtidas diretamente da matriz $[H] = [-P^T \vdots I_{n-k}]$, como mostrado abaixo. Seja $\underline{u} = (u_1,\ldots,u_n)$ uma palavra código correspondente a mensagem $\underline{m} = (m_1\ldots,m_k)$, o que implica em $u_i = m_i$, $i = 1,2,\ldots,k$.

Desde que $\underline{u}[H]^T = 0$, segue-se que

$$u_{k+j} = P_{1j}\, u_1 \oplus \ldots \oplus P_{kj}\, u_k \text{ , i.e.,}$$

$$u_{k+j} = P_{1j} m_1 \oplus \ldots \oplus p_{kj} m_k \quad \text{para} \quad j=1,2,\ldots,n-k \qquad (1\text{-}12)$$

que são as equações já encontradas em (1-8).

1.2.5 - Síndrome

Suponha que uma palavra código $\underline{u}$ de um código de bloco linear $C(n,k,d)$ com matriz geradora [G] e matriz de verificação de paridade [H] é transmitida através de um canal ruidoso. No receptor, uma énupla $\underline{r}$ é recebida, a qual pode diferir de $\underline{u}$ devido ao ruído adicionado durante a transmissão. A tarefa do decodificador é recuperar $\underline{u}$ a partir de $\underline{r}$.

Definição 1.2.5 - O vetor de $n-k$ componentes $\underline{S} = \underline{r}[H]^T$ é chamado de síndrome da énupla $\underline{r}$.

Como visto na seção anterior, $\underline{u}$ é uma palavra código se e somente se sua síndrome é nula.

Portanto, a síndrome de um vetor recebido na saída do canal pode ser usada para deteção/correção de erros.

O processo de decodificação envolve uma decisão sobre qual foi a palavra código transmitida. O arranjo padrão |24| suge-

re uma maneira de como fazer isto. No espaço V_n, as 2^n énuplas ' são distribuídas em 2^k conjuntos distintos, de modo que cada um ' deles contenha apenas uma palavra código. Todos os elementos de um dado conjunto possuem a mesma síndrome, e dois conjuntos distintos são associados a síndromes diferentes. Desta forma é estabelecida ' uma correspondência um a um entre uma configuração de erros (coset leader) e uma síndrome. Neste caso, o decodificador corrige erros ' quando a configuração de erros do canal é um "coset leader". Estes fatos implicam em um procedimento geral para decodificação de códigos de bloco lineares, denominado de busca sistemática |30|.

1.2.6 - Códigos Cíclicos Binários

Uma considerável parte das pesquisas em códigos de bloco estão concentradas numa subclasse conhecida como códigos cíclicos, introduzidos por Prange |26| em 1957.

Estes códigos são também os mais importantes do ponto de vista de aplicações em engenharia. Tudo isto é devido a sua forte estrutura matemática que permite uma considerável simplificação' na implementação dos sistemas. Os procedimentos de decodificação de códigos de blocos lineares são também aplicáveis a códigos cíclicos, todavia, propriedades algébricas decorrentes da estrutura cíclica ' permitem simplificações importantes, visto que a codificação e o cálculo da síndrome podem ser facilmente realizados empregando registros a deslocamento com conecções de realimentação. Sua estrutura inerentemente algébrica também torna possível encontrar vários métodos de decodificação simples e eficientes. Entre os procedimentos de decodificação mais importantes para este tipo de código, são conhecidos DECODIFICAÇÃO DE MEGGITT |36|, DECODIFICAÇÃO POR LÓGICA' MAJORITÁRIA |28| e o método conhecido como "ERROR-TRAPPING" |24| , os quais são de implementação simples e bastante atrativos.

Definição 1.2.6 - Um código de bloco é cíclico quando uma permutação cíclica, aplicada a qualquer de suas palavras, resulta ainda em uma palavra código, i.e., se $\underline{v} = (v_0, v_1, \ldots, v_{n-1})$ é uma palavra código, então $\underline{v}^i = (v_{n-i+1}, \ldots, v_0, v_1, \ldots, v_{n-i-1})$ também o é, considerando os índices reduzidos módulo n.

É possível tratar as componentes do vetor $\underline{v}$ como coeficientes de um polinômio de grau sempre menor ou igual a n-1, através de uma correspondência um a um:

$$\underline{v} = (v_0, \ldots, v_{n-1}) \Leftrightarrow v(x) = v_0 + v_1 x + \ldots + v_{n-1} x^{n-1} \qquad (1\text{-}13)$$

Fazendo uso de propriedades de campos finitos, prova-se [36] que todas as palavras de um código cíclico (n,k,d) são múltiplas de um polinômio g(X), de grau n-k, bem definido; e reciprocamente que todo polinômio de grau igual ou menor que n-1, e que seja divisível por g(X), é uma palavra código.

O polinômio g(X) é chamado polinômio gerador do código cíclico e é fator de X^n+1, i.e.,

$$g(X)h(X) = X^n+1, \quad \text{ou} \quad g(X)h(X) = 0 \mod X^n+1.$$

A representação de um código cíclico pode ser feita através do seu polinômio gerador g(X), da mesma forma que a matriz [G] representa um código de bloco linear.

Conforme mencionado, as palavras código são múltiplas do polinômio g(X). Desta forma, os polinômios $g(X), Xg(X), \ldots, X^{k-1}g(X)$ são palavras código e como são linearmente independentes, podem ser usadas como base para formar a matriz geradora do código cíclico, como mostrado a seguir.

$$[G] = \begin{vmatrix} X^{n-1} g(X) \\ \vdots \\ X\, g(X) \\ g(X) \end{vmatrix}$$

Para fins de codificação, a propriedade do deslocamento cíclico permite uma implementação sequencial da matriz [G].

Representando a mensagem a ser codificada por um polinômio $m(X)$ de grau menor ou igual a $k-1$, um codificador na forma sistemática gera a palavra código de acordo com

$$V(X) = C(X) + X^{n-k}\, m(X) \qquad (1\text{-}14)$$

onde $C(X)$ é o resto da divisão de $X^{n-k}\, m(X)$ por $g(X)$,

$$C(X) = \mathrm{rem}\{X^{n-k}\, m(X)/g(X)\}. \qquad (1\text{-}15)$$

A implementação do codificador pode ser feita através de registro a deslocamento com $n-k$ estágios, pois multiplicações e divisões de polinômios com coeficientes em GF(2) podem ser efetuadas facilmente com o auxílio de registradores a deslocamento |11|.

Um codificador com k estágios ao invés de $n-k$ estágios pode também ser implementado |30|, levando em consideração o polinômio de verificação de paridade $h(X)$.

A síndrome pode ser facilmente determinada, conforme será mostrado a seguir. De um modo geral, a palavra recebida pode ser representada pela expressão $r(X) = V(X) + e(X)$, onde $e(X)$ é um polinômio correspondente à configuração de erros introduzida no canal. A síndrome pode ser determinada como o resto da divisão entre o polinômio recebido e o polinômio gerador, i.e.,

$$S(X) = \mathrm{rem}\{r(X)/g(X)\} = \mathrm{rem}\left\{\frac{V(X)}{g(X)} + \frac{e(X)}{g(X)}\right\} \qquad (1\text{-}16)$$

Como $V(X) = m(X)\ g(X)$, segue-se que

$$S(X) = \text{rem}\ \{e(X)/g(X)\} \qquad (1\text{-}17)$$

Caso $S(X)$ seja zero, o decodificador aceita a palavra recebida como pertencente ao código. No caso contrário, i.e., $S(X) \neq 0$, considera que ocorreram erros durante a transmissão. Desta maneira é fácil perceber a simplicidade de um circuito decodificador para deteção de erros quando são empregados códigos cíclicos.

1.2.7 - Códigos de Hamming e Códigos B.C.H

Os códigos de Hamming, introduzidos em 1950 [16], foram os primeiros códigos não trivais propostos para corrigir erros. São códigos lineares que corrigem um erro por bloco e têm distância mínima igual a 3. Para cada inteiro $c>1$, existe um código de Hamming com os seguintes parâmetros:

$$n = 2^c-1 \qquad k = 2^c-1-c \qquad e \qquad d = 3.$$

A condição $n = 2^c-1$ é escolhida de modo a assegurar disponibilidade de redundância suficiente para verificar a ocorrência de um erro por palavra, pois o número de síndromes não nulas 2^c-1 resulta exatamente igual ao número de posições n em que o erro pode estar localizado. Isto significa que os códigos de Hamming são códigos perfeitos conforme a definição abaixo.

Definição 1.2.7 - Um código de bloco (n,k,d) corretor de t erros e definido em $GF(q)$ é perfeito se e somente se

$$\sum_{i=0}^{t} (q-1)^i\ C_n^i = q^{n-k}$$

Com excessão dos códigos de Hamming, o código de Golay (23,12,7) com $t=3$ e o código ternário (11,6,7) com $t=3$, não existem outros códigos perfeitos não triviais.

É interessante observar que os códigos de Hamming são também códigos cíclicos e que a taxa (eficiência) é dada por $R = 1 - \frac{c}{2^c-1}$, consequentemente são códigos cuja eficiência tende a 1 assintóticamente. Os códigos (7,4,3), (15,11,3) e (31,26,3) serão utilizados nas análises realizadas nos capítulos seguintes.

Com relação aos códigos BCH, estes códigos foram criados independentemente por Hocquenghen (1959) e por Bose-Chandhuri (1960); são códigos cíclicos, e representam a classe mais importante de códigos de bloco com algorítmos algébricos de decodificação. Para quaisquer inteiros positivos m e t $(t<2^m-1)$ existe um código BCH com os seguintes parâmetros:

$$n = 2^m-1 \qquad n-k \leq mt \qquad e \qquad d \geq 2t + 1.$$

O teorema fundamental dos códigos BCH é a seguir enunciado.

Teorema 1.1 - O código BCH cujo polinômio gerador tem $d-1$ raízes consecutivas $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{d-1}$, tem distância mínima pelo menos d.

Para um tratamento completo destes códigos o leitor deve consultar as referências |3| e |26|.

1.3 - CÓDIGOS CONVOLUCIONAIS

1.3.1 - Considerações Gerais

Na classe de códigos denominada de códigos convolucionais, a redundância adicionada a cada bloco é função de mais de um bloco de informação, de modo que estes não são tratados de forma in

dependente como nos códigos de bloco. Estes códigos foram primeiramente introduzidos |26| por P.Elias (1955), e podem ser classificados como fixos ou variantes no tempo. Entretanto, a grande maioria dos códigos convolucionais empregados é invariante no tempo, de forma que apenas estes serão abordados aqui.

Na decodificação de códigos convolucionais, vários sub-blocos sucessivos recebidos são processados. Idealmente, toda sequência transmitida deveria ser empregada na decodificação, o que não é feito devido a limitações práticas e por introduzir um retardo muito grande na decodificação. Todavia, a degradação resultante desta simplificação é inteiramente aceitável.

Num sentido restrito, códigos convolucionais podem ser vistos como uma classe especial de códigos de bloco lineares, porém, uma observação mais cuidadosa revela que a estrutura convolucional proporciona a um código linear propriedades superiores que melhoram o desempenho.

A estrutura de um código convolucional pode ser exibida por meio de diversas representações, visto que um codificador convolucional fixo pode ser considerado como uma máquina sequencial de estados finitos, invariante no tempo.

1.3.2 - O Codificador Convolucional

Na entrada do codificador é alimentado um bloco com k símbolos de informação (elementos de GF(q)), e um bloco com $n>k$ símbolos código é gerado e aparece na saída do referido codificador. Os símbolos de saída são combinações lineares em GF(q) dos símbolos de entrada dos $N-1$ blocos precedentes. Deste modo, os n dígitos de saída dependem não somente dos k dígitos de mensagem de um mesmo sub-bloco, mas também dos $N-1$ sub-blocos de mensagem anteriores.

Um código assim gerado é dito ser um código convolucional (n,k,N), com comprimento de restrição no codificador de N blocos. A taxa assintótica para este código é $R = k/n$, e em geral, k e n são inteiros pequenos.

A formulação matricial para estes códigos é descrita a seguir:

Inicialmente é considerado um conjunto de $k.(n-k)$ vetores com N dígitos binários cada, denominados SUBGERADORES:

$$g(i,j) = (g_0(i,j),\ g_1(i,j),\ldots,g_{N-1}(i,j)) \qquad (1\text{-}18)$$

Para $i = 1,2,\ldots,k$ e $j = 1,2,\ldots,n-k$.

A partir deste conjunto, forma-se uma matriz com k linhas,

$$\begin{vmatrix} g_\infty(1) \\ \vdots \\ g_\infty(k) \end{vmatrix} = [I_k\ P_0\ 0\ P_1\ 0\ P_2 \ldots 0\ P_{N-1}\ 0^{\text{zeros}_{\to\infty}}] \qquad (1\text{-}19)$$

Onde I_k tem dimensão $k\text{x}k$

0 tem dimensão $k\text{x}k$

P_ℓ tem dimensão $k\text{x}(n-k)$, $0 \leq \ell \leq N-1$

e

$$[P_\ell] = \begin{vmatrix} g_\ell(1,1) & g_\ell(1,2)\ldots g_\ell(1,n-k) \\ \vdots & \vdots \qquad\qquad \vdots \\ g_\ell(k,1) & g_\ell(k,2)\ldots g_\ell(k,n-k) \end{vmatrix} \qquad (1\text{-}20)$$

Assim, cada linha $g_\infty(i)$ é um vetor semi-infinito com componentes não nulas confinadas aos primeiros nN dígitos.

O vetor $g(i)$ contendo somente as primeiras nN posições do vetor $g_\infty(i)$ é dito um GERADOR.

Um código convolucional fixo linear pode ser representado por uma matriz geradora sob a forma:

$$G_\infty = \begin{vmatrix} I_k & P_0 & 0 & P_1 & 0 \ldots 0 & P_{N-1} & \text{zeros}_{\to\infty} & & \\ & I_k & P_0 & 0 \ldots\ldots\ldots 0 & P_{N-1} & \text{zeros}_{\to\infty} & & \\ & & I_k & P_0 \ldots\ldots\ldots\ldots & 0 & P_{N-1} & \text{zeros}_{\to\infty} & \\ & & & & \downarrow \infty & & & \end{vmatrix} \quad (1\text{-}21)$$

Alguns exemplos apresentados esclarecem como determinar a matriz G_∞ a partir dos subgeradores.

<u>Exemplo 1.3</u> - Considere um código convolucional sistemático (2,1,4) o qual é gerado de acordo com o subgerador $g(1,1) = (1\ 1\ 0\ 1)$.

O gerador é $g(1) = (1\ 1\ 0\ 1\ 0\ 0\ 0\ 1)$ e a matriz geradora é determinada como sendo

$$G_\infty = \begin{vmatrix} 1\ 1 & 0\ 1 & 0\ 0 & 0\ 1 & & & \\ & 1\ 1 & 0\ 1 & 0\ 0 & 0\ 1 & & \\ & & 1\ 1 & 0\ 1 & 0\ 0 & 0\ 1 & \\ & & & & \downarrow \infty & & \end{vmatrix}$$

Portanto, para uma mensagem a ser codificada $\underline{m} = (\underline{1}\ \underline{0}\ \underline{0}\ \underline{1}\ \underline{1} \ldots)$, a sequência codificada é encontrada por $\underline{c} = \underline{m}\ G_\infty$, i.e., $\underline{C} = (\underline{11}\ \underline{01}\ \underline{00}\ \underline{10}\ \underline{10} \ldots)$.

<u>Exemplo 1.4</u> - Considere agora o código convolucional (4,3,3) com os seguintes subgeradores:

$$g(1,1) = (1\ 1\ 1)$$
$$g(2,1) = (1\ 0\ 1)$$
$$g(3,1) = (1\ 1\ 0)$$

Os geradores são determinados como sendo:

$$g(1) = (1\ 0\ 0\ 1 \quad 0\ 0\ 0\ 1 \quad 0\ 0\ 0\ 1)$$

$$g(2) = (0\ 1\ 0\ 1 \quad 0\ 0\ 0\ 0 \quad 0\ 0\ 0\ 1)$$

$$g(3) = (0\ 0\ 1\ 1 \quad 0\ 0\ 0\ 1 \quad 0\ 0\ 0\ 0)$$

e a matriz geradora será dada por

$$G_\infty = \left|\begin{array}{cccccccccccc}
1\ 0\ 0 & 1 & 0\ 0\ 0 & 1 & 0\ 0\ 0 & 1 & & & & & & \\
 & \vdots & & \vdots & & \vdots & & & & & & \\
0\ 1\ 0 & 1 & 0\ 0\ 0 & 0 & 0\ 0\ 0 & 1 & & & & & & \text{zeros} \\
 & \vdots & & \vdots & & \vdots & & & & & & \rightarrow \infty \\
0\ 0\ 1 & 1 & 0\ 0\ 0 & 1 & 0\ 0\ 0 & 0 & & & & & & \\
 & & & & & & & & & & & \\
 & & 1\ 0\ 0 & 1 & 0\ 0\ 0 & 1 & 0\ 0\ 0 & 1 & & & & \\
 & & & \vdots & & \vdots & & \vdots & & & & \\
 & & 0\ 1\ 0 & 1 & 0\ 0\ 0 & 0 & 0\ 0\ 0 & 1 & & & & \text{zeros} \\
 & & & \vdots & & \vdots & & \vdots & & & & \rightarrow \infty \\
 & & 0\ 0\ 1 & 1 & 0\ 0\ 0 & 1 & 0\ 0\ 0 & 0 & & & & \\
 & & & & & \downarrow & & & & & & \\
 & & & & & \infty & & & & & &
\end{array}\right|$$

Deve ser observado que $P_0^T = (1\ 1\ 1)$, $P_1^T = (1\ 0\ 1)$, e $P_2^T = (1\ 1\ 0)$.

Um código convolucional (n,k,N) pode também ser especificado pela sua matriz de verificação de paridade H_∞, tal como um código de bloco. Novamente é verificada a relação $G_\infty H_\infty^T = 0$, donde segue-se que $\underline{c}$ é uma sequência código se e somente se $\underline{c}\ H_\infty^T = \underline{0}$.

Para um código convolucional sistemático, ou seja, com matriz geradora da forma descrita em (1-21), a matriz de verificação de paridade é dada por

$$H_\infty = \left|\begin{array}{ccccccccccccc}
P_0^T & I_{n-k} & & & & & & & & & & & \\
P_1^T & 0 & P_0^T & I_{n-k} & & & & & & & & & \\
P_2^T & 0 & P_1^T & 0 & P_0^T & I_{n-k} & & & & & & & \\
\vdots & \vdots & \vdots & \vdots & & & & & & & & & \\
P_{N-1}^T & 0 & P_{N-2}^T & \cdots & \cdots & \cdots & 0 & P_1^T & 0 & P_0^T & I_{n-k} & & \\
 & & P_{N-1}^T & 0 & P_{N-2}^T & \cdots & \cdots & \cdots & 0 & P_1^T & 0 & P_0^T & I_{n-k} \\
 & & & & & \downarrow & & & & & & & \rightarrow \infty \\
 & & & & & \infty & & & & & & &
\end{array}\right| \qquad (1\text{-}22)$$

Exemplo 1.5 - Como um exemplo, a matriz de verificação de paridade para o código (4,3,3) descrito no exemplo 1.4 é determinada abaixo.

$$H_\infty = \left|\begin{array}{ccccccccccccc} 1 & 1 & 1 & 1 & & & & & & & & & \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & & & & & \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & \\ & & & & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ & & & & & & & & & & & \downarrow & & & \rightarrow \infty \\ & & & & & & & & & & & \infty & & & \end{array}\right|$$

1.3.3 - Outra Representação da Matriz Geradora

É possível também representar a matriz G_∞ numa outra forma que facilita o projeto dos circuitos codificadores, como será mostrado adiante.

Para um código convolucional (n,k,N), a matriz G_∞ pode ser escrita sob a forma:

$$G_\infty = \left|\begin{array}{ccccc} g_0 & g_1 & g_2 \cdots & g_{N-1} & \\ & g_0 & g_1 \cdots\cdots & & g_{N-1} \\ & & g_0 \; g_1 \cdots\cdots & & g_{N-1} \\ & & & \downarrow & \rightarrow \infty \\ & & & \infty & \end{array}\right| \qquad (1\text{-}23)$$

Onde g_j, $j = 0,1,\ldots N-1$ são matrizes kxn.

Se o código é sistemático, comparando (1-21) e (1-23), é possível observar que existem relações da forma

$$g_0 = I_k \, P_0 \; , \; g_1 = 0 \; P_1 \; , g_2 = 0 \; P_2 , \ldots , g_{N-1} = 0 \; P_{N-1} \; .$$

A matriz de verificação de paridade também pode ser es-

crita na forma abaixo,

$$H_\infty = \begin{vmatrix} h_0 & & & & & \\ h_1 & h_0 & & & & \\ h_2 & h_1 & h_0 & & & \\ \cdots & \cdots & \cdots & & & \\ h_{N-1} & h_{N-2} & \cdots & h_1 & h_0 & \\ & h_{N-1} & h_{N-2} & \cdots & h_1 & h_0 \\ & & & \downarrow \infty & & \rightarrow \infty \end{vmatrix} \qquad (1\text{-}24)$$

Onde $h_0 = P_0^T\ I_{n-k}$, $h_1 = P_1^T\ 0$, $h_2 = P_2^T\ 0$, ... , $h_{N-1} = P_{N-1}^T\ 0$.

Os k geradores podem ser facilmente obtidos a partir' da relação

$$\begin{vmatrix} g(1) \\ \vdots \\ g(k) \end{vmatrix} = \begin{vmatrix} g_0 & g_1 & \cdots & g_{N-1} \end{vmatrix} \qquad (1\text{-}25)$$

Os exemplos apresentados abaixo mostram como esta representação é refletida na implementação do circuito codificador.

<u>Exemplo 1.6</u> - Um circuito codificador para o código convolucional' (2,1,3) especificado através de g_0, g_1 e g_2 abaixo, é mostrado na figura 1.1.

$g_0 = (1,1)$

$g_1 = (1,0)$ => gerador $g(1) = (1\ 1\ \ 1\ 0\ \ 1\ 1)$

$g_2 = (1,1)$

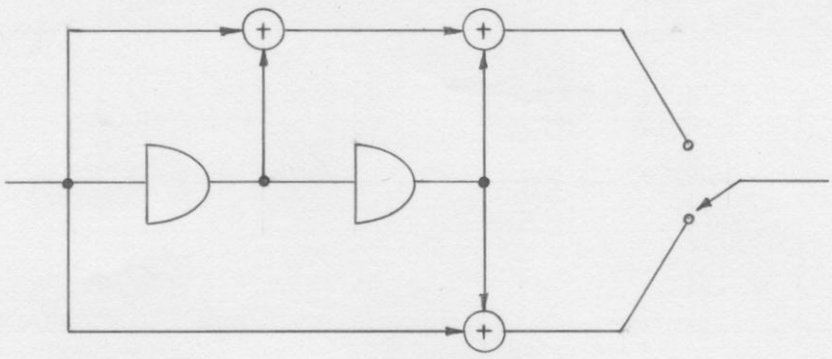


Figura 1.1 - CODIFICADOR PARA O CÓDIGO CONVOLUCIONAL (2,1,3).

<u>Exemplo 1.7</u> - Um circuito codificador para o código convolucional' não sistemático (3,2,2) especificado através das matrizes g_0 e g_1' abaixo, está apresentado na figura 1.2.

$$g_0 = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

$$g_1 = \begin{vmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix}$$

=> geradores

$g(1) = (1\ 1\ 1\ 1\ 0\ 1)$

$g(2) = (0\ 1\ 0\ 1\ 1\ 0)$

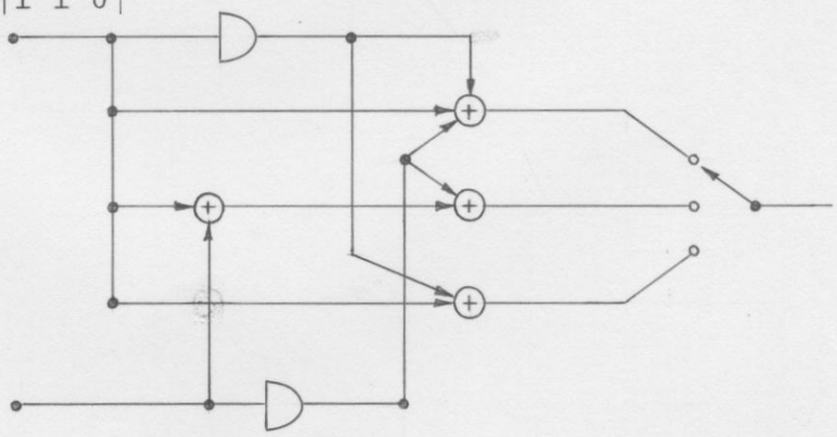


Figura 1.2 - CODIFICADOR PARA O CÓDIGO CONVOLUCIONAL (3,2,2).

Portanto, para implementação do codificador de um código convolucional (n,k,N) são necessários k.(N-1) estágios de registro a deslocamento com ligações determinadas pelos geradores.

1.3.4 - Diagrama em Árvore, Treliça e Diagrama de Estado

Uma das maneiras de representar a sequência de saída resultante da codificação de uma dada sequência de entrada para um código convolucional fixo, é por meio de uma representação da árvore' associada ao codificador.

Os bits de entrada do codificador são indicados pelo caminho seguido na árvore, enquanto que os bits de saída são indicados nos ramos. Um zero de entrada é representado pelo ramo superior de uma bifurcação, enquanto que o um é representado pelo ramo inferior.

O diagrama em árvore para um código convolucional não sistemático (4,1,3) é apresentado na figura 1.3 abaixo.

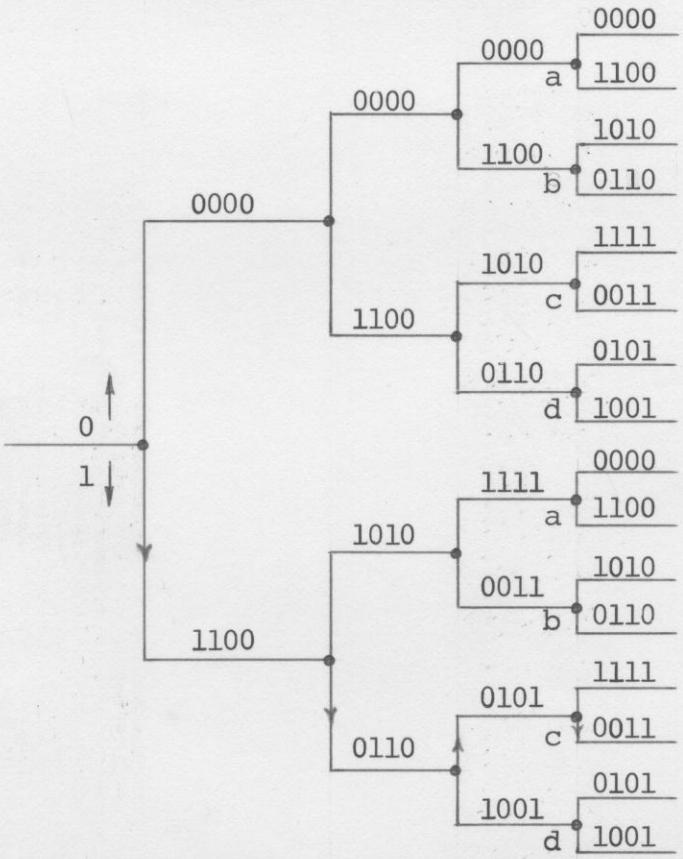


Figura.1.3 - DIAGRAMA EM ÁRVORE PARA O CÓDIGO CONVOLUCIONAL (4,1,3).

Deste modo, a sequência 1 1 0 1 0 ... aplicada na entrada do codificador produz uma sequência de saída

1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 0 1 1 0 ...

Observando-se o diagrama de árvore, nota-se que sua estrutura torna-se repetitiva após um certo número de ramos, o que está relacionado com o fato do codificador convolucional ser uma máquina com estados finitos. Isto torna possível fazer uma modificação que consiste em juntar os nós indicados por uma mesma letra, fazendo com que o diagrama em árvore evolua para um diagrama de treliça.

A treliça para o código (4,1,3) representado na figura 1.3 é facilmente obtida, resultando no diagrama mostrado abaixo.

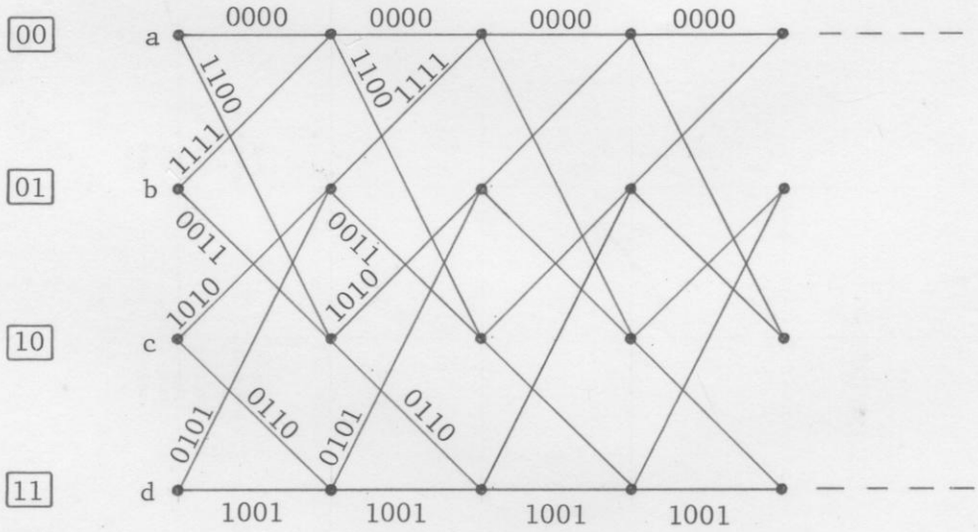


Figura 1.4 - DIAGRAMA DE TRELIÇA PARA CÓDIGO CONVOLUCIONAL (4,1,3).

Pode ainda ser observado que é possível fazer uma representação em diagrama de estado de um codificador convolucional. A partir da treliça é possível construir este diagrama, que representa perfeitamente o codificador. O diagrama para o exemplo em questão é mostrado na figura 1.5.

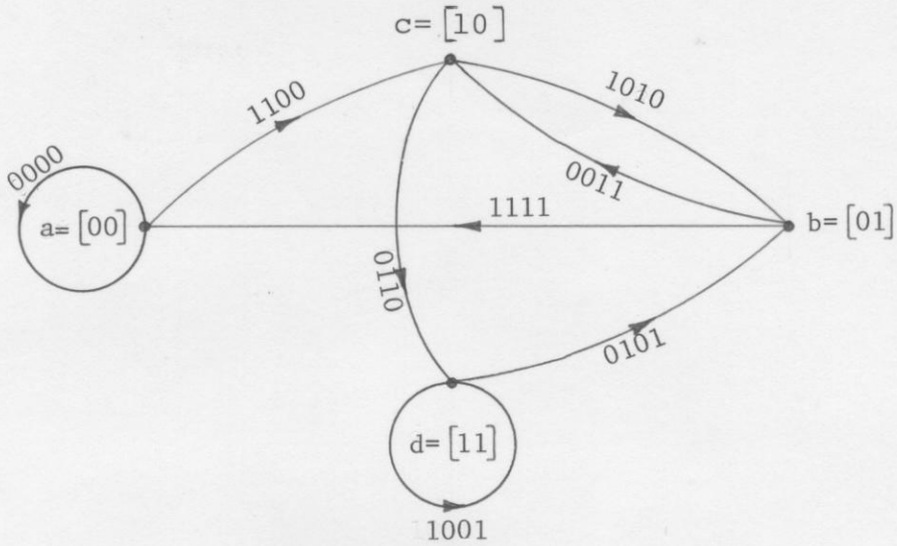


Figura 1.5 - DIAGRAMA DE ESTADO PARA O CÓDIGO CONVOLUCIONAL (4,1,3).

Utilizando este diagrama é possível a determinação do conjunto de pesos dos caminhos do código de uma forma relativamente simples |37|.

A determinação da distância livre para um código convolucional, a qual é obtida a partir do conjunto de distâncias entre o caminho todo zero e os demais caminhos que dele divergem, também pode ser feita através do diagrama de estado |30|.

1.4 - DECODIFICAÇÃO DE CÓDIGOS CONVOLUCIONAIS

Vários procedimentos são usuais na decodificação de códigos convolucionais. Entre as técnicas mais comumente utilizadas podem ser citadas a DECODIFICAÇÃO SEQUENCIAL |40|, (algorítmo de FANO/WOZENCRAFT) e o ALGORÍTMO DE VITERBI |37|, entre outras.

Para alguns códigos, a decodificação também pode ser realizada utilizando LÓGICA MAJORITÁRIA |36|.

O leitor interessado nestes procedimentos deve se diri-

gir à literatura apropriada |1|,|28|,|40|.

Ainda com relação a decodificação de códigos convolucionais, serão apresentados no capítulo III, duas maneiras de utilizar técnicas de decisão suave. Inicialmente, um decodificador será apresentado, o qual utiliza decisão suave e é ótimo no sentido de que minimiza a probabilidade de erro por bit transmitido. É também conhecido que o algorítmo de Viterbi para decodificação de códigos convolucionais maximiza a probabilidade a posteriori das sequências código em canais sem memória.

CAPÍTULO II

DECODIFICAÇÃO USANDO DECISÃO SUAVE

2.1 - INTRODUÇÃO

Como já discutido no capítulo anterior, sabe-se que na prática, empregam-se largamente algorítmos de decodificação baseados em propriedades estruturais de códigos algébricos, o que permite frequentemente uma implementação relativamente simples em termos de circuitos digitais. Deve ser observado, entretanto, que estes procedimentos apresentam uma séria deficiência, uma vez que assumem um modelo de canal no qual em cada período de tempo um dos q possíveis sinais do alfabeto é transmitido e um deles é recebido.

Este modelo implica em um receptor que efetua uma decisão abrupta, baseada no sinal recebido, escolhendo das q letras do alfabeto do transmissor aquela que foi a mais provavelmente transmitida. Tal receptor descarta, portanto, toda informação acerca da confiabilidade desta escolha, bem como acerca das probabilidades das outras letras que não a escolhida. Este fato torna bastante claro que o desempenho dos sistemas codificados pode ser melhorado pelo uso da informação probabilística associada à palavra recebida.

Numa recepção que utiliza decisão abrupta, esta informação é perdida na quantização que precede a decodificação. Um receptor que emprega decisão suave é aquele que tenta reter toda (ou parte) da informação probabilística para uso apropriado pelo decodificador.

Em geral, os códigos corretores de erros empregados em receptores práticos são códigos lineares binários. A grande parte dos procedimentos existentes para decodificação deste tipo de código assumem a existência de canais com saída binária, ou seja, empregam um quantizador antes da decodificação para extrair a informação na forma digital. A maioria dos detetores existentes constam basicamente de um demodulador analógico, seguido por um decodificador digital que opera sobre os dígitos binários produzidos pelo demodulador. O bloco demodulador atua como um quantizador que apresenta um limiar de decisão, resultando em duas saídas distintas. Isto significa que de acordo com o valor da amostra que é colhida (acima ou abaixo do limiar de referência),um dígito binário correspondente (1 ou 0) é entregue ao decodificador. É desta forma que a informação probabilística, potencialmente útil, entregue ao demodulador é inteiramente destruída, fato que reduz substancialmente a eficiência do sistema de comunicação. Um processo de deteção (demodulação e decodificação) com decisão suave utiliza a informação de natureza probabilística associada à mensagem recebida, de modo a obter uma melhor estimativa da mesma. A idéia é fornecer ao decodificador informação sobre a confiabilidade dos dígitos recebidos, evitando assim a degradação que resulta quando se utiliza antes da decodificação uma quantização de apenas duas regiões.

A maioria da perda de informação pode ser recuperada se a saída do demodulador é quantizada em 2^Q regiões, 2^{Q-1} em cada lado do limiar de referência.Então,para cada dígito de entrada modulado,o circuito demodulador fornece um dígito de decisão (indicando se a saída está acima ou abaixo do limiar),e Q-1 dígitos de confiabilidade(indicando quão distante a saída está do limiar de decisão). Os Q dígitos de decisão suave constituem então a saída total,ao invés de somente um dígito,como no caso de decisão abrupta.

Na figura 2.1 é mostrado um exemplo para Q=2.

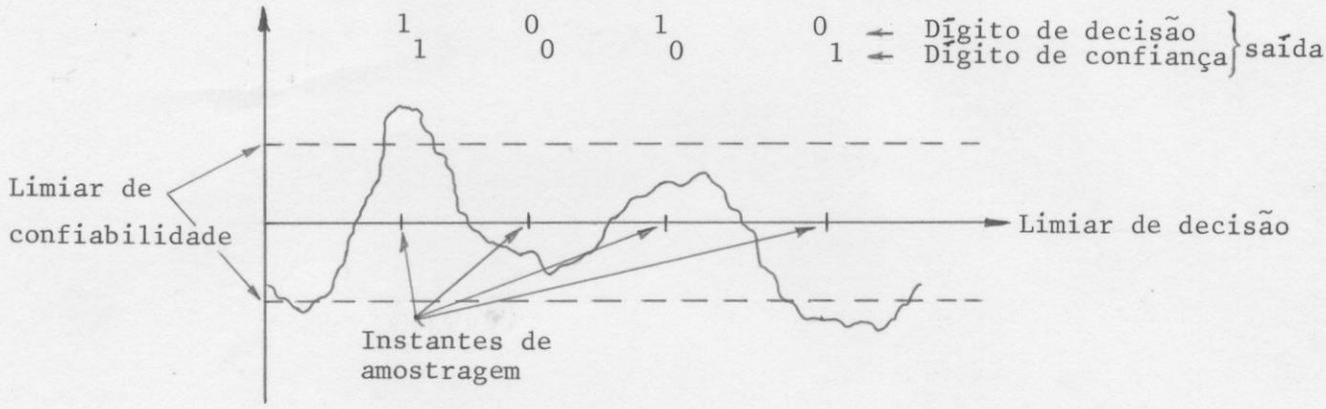


Figura 2.1 - DECISÃO SUAVE - QUANTIZAÇÃO EM 4 REGIÕES DE CONFIABILIDADE.

A mais simples forma de decisão suave é o apagamento, ou deteção por zona nula. Para um estudo mais detalhado, as referências |4| e |5| são apropriadas. Neste método, a região próxima (imediatamente acima e imediatamente abaixo) do limiar de decisão é chamada de zona nula. Saídas caindo dentro desta região são entregues ao decodificador rotuladas como apagamento E, como observado no exemplo da figura 2.2.

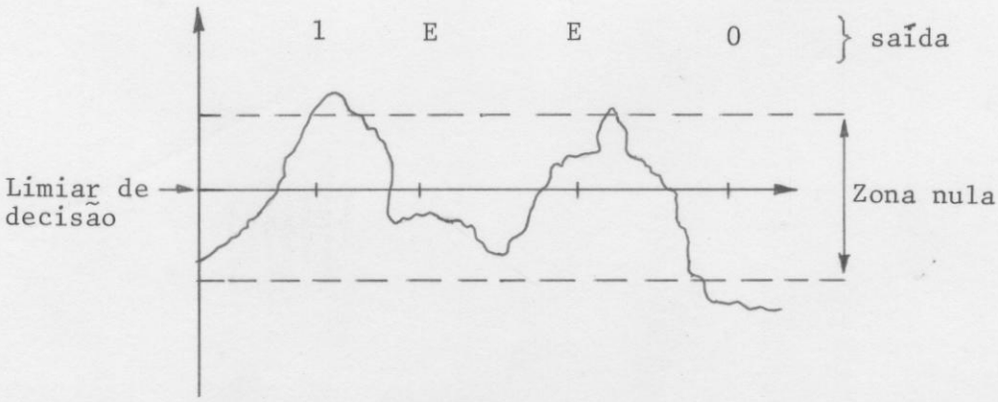


Figura 2.2 - DETEÇÃO POR ZONA NULA.

Quando o decodificador utiliza decisão abrupta, é assumido um modelo de canal BSC, enquanto que o uso de apagamento resulta em um canal BEC. Os algorítmos algébricos de decodificação podem ser modificados para manipularem eficientemente os apagamentos.

O decodificador utilizando deteção por zona nula tem agora algum conhecimento de onde os erros provavelmente ocorreram, e pode decodificar de acordo com este fato, ou seja, os d-1 erros detetados são localizados através da informação de confiabilidade.

Desta maneira, a potência de correção de erros de um código pode ser aproximadamente dobrada, desde que um código com distância mínima d pode corrigir d-1 apagamentos, mas somente $\left\lfloor \frac{d-1}{2} \right\rfloor$ erros.

A utilização de mais de duas regiões de quantização é uma tentativa de diminuir a perda de informação que resulta quando uma decisão abrupta é realizada.

Portanto, o detetor ótimo, i.e., aquele que no processo de decisão retém toda a informação contida no sinal recebido, seria aquele que entregaria ao decodificador o valor exato das amostras colhidas. Mais tarde, no capítulo III, estes detetores serão abordados.

Uma maneira de se utilizar os dígitos de confiabilidade no processo de decodificação é através do conceito de distância suave.

A distância suave d_s entre duas palavras código pode ser calculada de acordo com o procedimento descrito abaixo:

1) - invertendo os dígitos de confiança da palavra se seu dígito de informação correspondente é zero.

2) - Somando módulo 2 as duas palavras modificadas de acordo com o

passo 1.

3) - Dividindo a sequência resultante em n subsequências de Q dígitos cada.

4) - Interpretando cada uma das subsequências de Q dígitos como um número binário e convertendo para decimal.

5) - Somando os números obtidos em 4, resultando em um decimal total.

Assim, d_s entre 0 0 0 1 1 1 e 0 1 1 0 1 1 é obtida por

$$
\begin{array}{r}
0\,1\,1 \;\vdots\; 1\,1\,1 \\
\oplus \quad \vdots \\
0\,0\,0 \;\vdots\; 0\,0\,0 \\
\hline
0\,1\,1 \;\vdots\; 1\,1\,1 \\
3 \quad + \quad 7 \quad = 10
\end{array}
$$

O uso de distância suave no processo de decodificação será ilustrado a seguir.

No exemplo abaixo é assumido Q = 3 e um código com exatamente duas palavras, 0 ou 1, para o qual d=1 e R=1.

A figura 2.3 mostra as possíveis saídas obtidas após o demodulador.

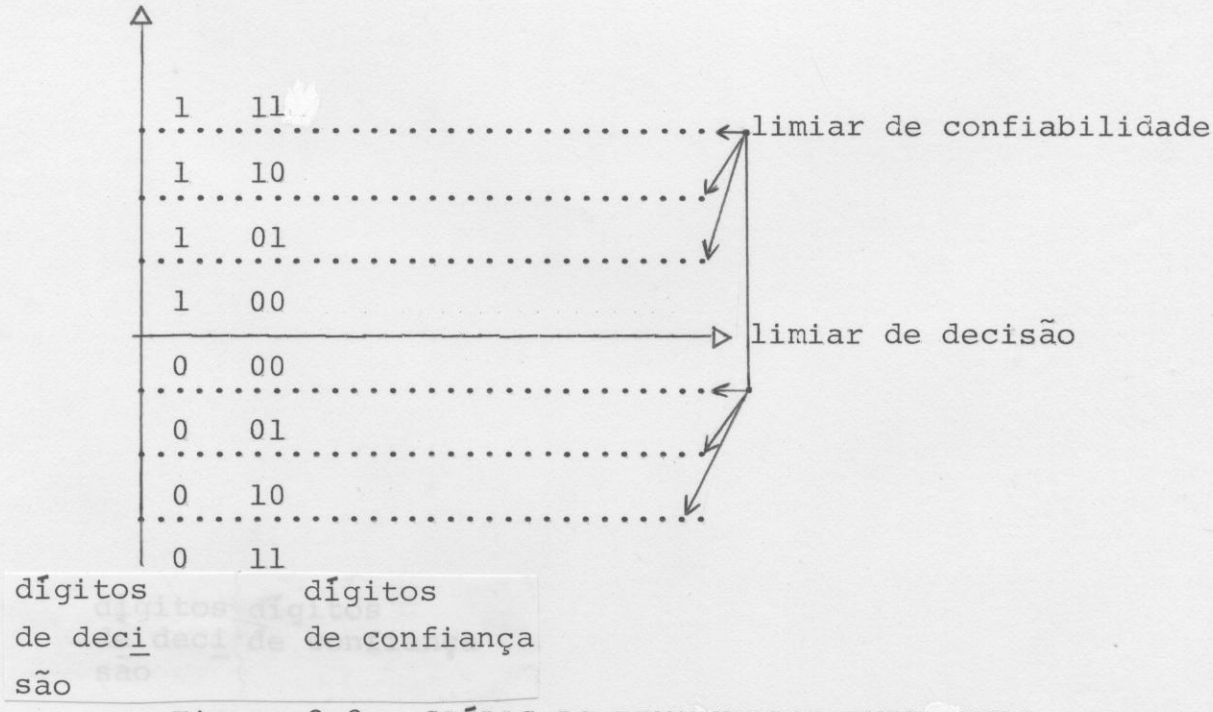


Figura 2.3 - SAÍDAS DO DEMODULADOR EMPREGANDO DECISÃO SUAVE.

É admitido que a palavra 1 é transmitida, e que devido ao ruído do canal, 0 0 0 é a saída do demodulador. As duas possibilidades de plena confiança para a saída do demodulador são 0 1 1 e 1 1 1, de maneira que a distância de decisão suave d_s, para cada caso é

$$d_s(0\ 0\ 0\ ,\ 0\ 1\ 1) = 3$$
$$d_s(0\ 0\ 0\ ,\ 1\ 1\ 1) = 4.$$

Desta maneira, 0 será selecionado incorretamente como tendo sido a palavra transmitida, quando é empregado um decodificador por distância suave mínima.

Agora, será considerado um novo exemplo no qual se assume as palavras código como sendo 0 0 e 1 1, resultando em um código com d=2 e R=0,5. É admitido que a palavra 1 1 é transmitida, mas 0 0 0 1 1 1 é recebida na saída do demodulador, ou seja, o ruído afetou o primeiro dígito da palavra código.

As duas saídas de plena confiança são agora 0 1 1 0 1 1 e 1 1 1 1 1 1, de modo que

$$d_s(0\ 0\ 0\ 1\ 1\ 1,\ 0\ 1\ 1\ 0\ 1\ 1) = 10$$
$$d_s(0\ 0\ 0\ 1\ 1\ 1,\ 1\ 1\ 1\ 1\ 1\ 1) = 4.$$

Portanto, 1 1 será selecionada corretamente, e um único erro foi corrigido, embora o código possua d=2.

Este tipo de deteção com decisão suave é particularmente aplicável à decodificação por distância mínima (máximo de verossimilhança) para códigos de bloco, e também para códigos convolucionais utilizando o algorítmo de Viterbi ou decodificação sequencial.

2.2 - ALGUNS PROCEDIMENTOS SUBÓTIMOS

Nesta seção são descritos algorítmos de decodificação de códigos de bloco que empregam decisão suave e que são subótimos, no sentido que o ganho teórico máximo possível não é atingido. Em alguns casos isto se deve ao fato do uso de um número finito de regiões de quantização, de maneira a viabilizar a implementação prática.

Tais procedimentos, embora subótimos, representam uma melhoria considerável no desempenho com relação a sistemas que utilizam decisão abrupta.

2.2.1 - Decisão Suave na Decodificação de Códigos de um Único Dígito de Paridade

O procedimento descrito a seguir é devido a Farrell e Kalligeros |12|, onde se utilizou como exemplo um código de bloco (8,7,2) e um detetor com decisão suave quantizado em 8 regiões, cujos níveis são $\pm 0,25V$, $\pm 0,5V$, $\pm 0,75V$ para sinais de $\pm 1V$.

O decodificador deve operar de acordo com a seguinte sequência:

1) - avaliar os dígitos de decisão $k_1k_2 \ldots k_6k_7c_1$ e seus correspondentes "bytes" de confiabilidade.

2) - Recalcular o dígito de paridade c_1 a partir dos dígitos de informação e compará-lo com o valor recebido c_1.

   a) Se c_1 estiver correto, é assumido que não ocorreram erros, e a palavra é liberada.

   b) Em caso contrário, inverter o dígito de mais baixa confiabilidade, de modo a corrigir o erro isolado mais provável, e liberar então a palavra.

A curva de desempenho, obtida por simulação em computa

dor, é mostrada a seguir na figura 2.4 abaixo.

Para relações sinal/ruído maiores que -2,5dB observa-se um ganho em relação ao sistema sem codificação, o qual é superior ' àquele obtido por um código de Hamming de mesmo comprimento e/ou ' eficiência.

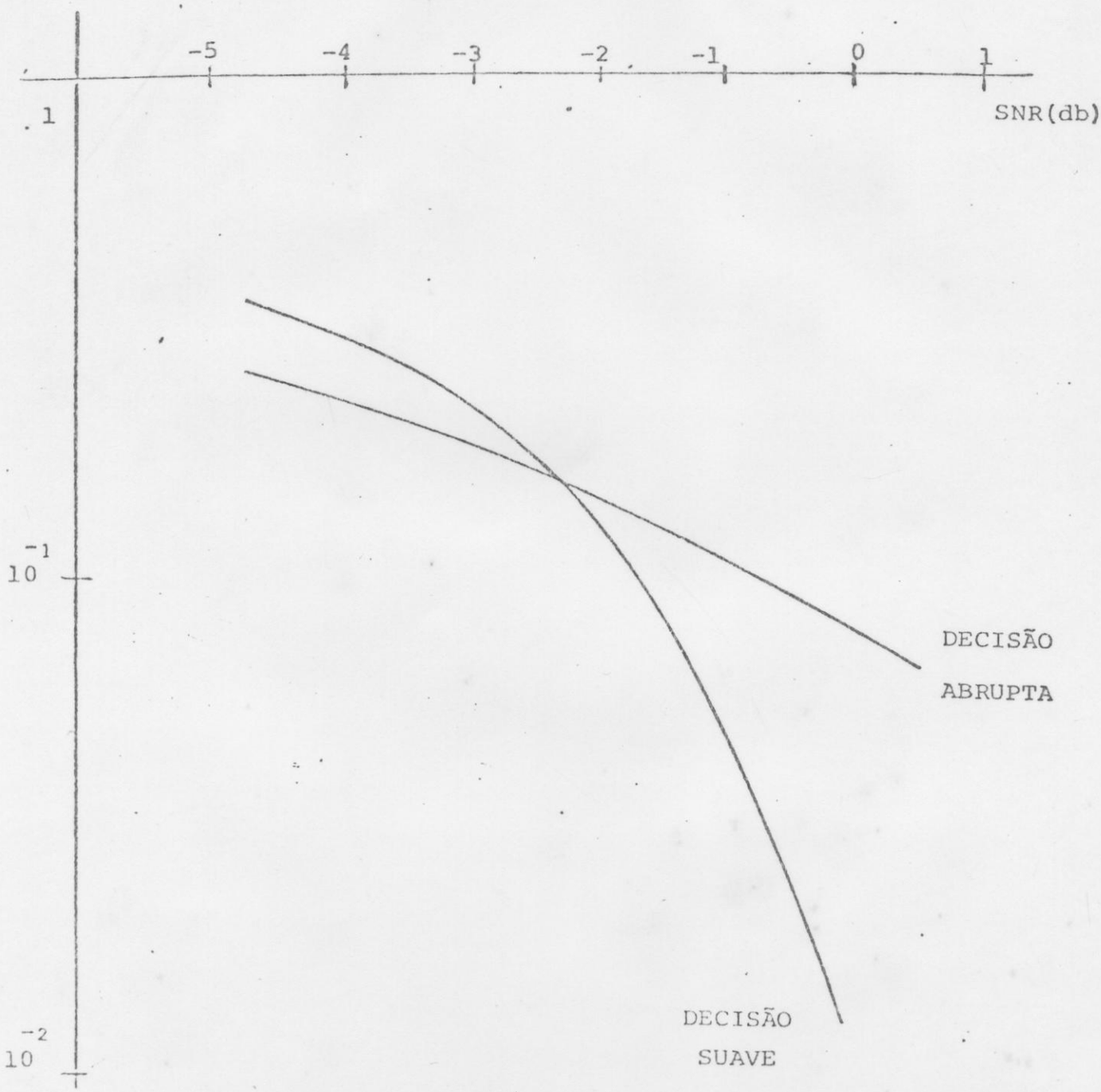


Figura 2.4 - CURVAS DE DESEMPENHO DO ALGORÍTMO DE FARRELL E KALLIGEROS.

2.2.2 - Decodificação por Distância Suave Mínima

Decodificação por distância mínima usando decisão suave foi empregada em um sistema estudado por Farrell e Munday (1976) |12|. É basicamente um sistema de comunicação em HF sem canal de retorno. As características do sistema são:

1) - baixa taxa de dados, aproximadamente 50 bits/seg.
2) - Um código produto |36| é empregado, com $n = 15x15 = 225$ e $d = 3x3 = 9$ codificado e decodificado em cascata.
3) - Intercalamento de bits para combater a mistura de erros aleatórios e "burst" do canal de HF.
4) - "Sequence inversion keying" |11| de aproximadamente 100 bits / seg, para espalhar o espectro num canal de HF com 3kHz. Para tal, uma sequência - m de 15 dígitos foi empregada.
5) - Modulação ASK, com deteção coerente empregando PLL.
6) - Correlação dos dígitos da sequência m para determinar se a sequência está invertida ou não (dígito de decisão) e o grau de correlação na presença de erros do canal (dígitos de confiança); o que é equivalente a um esquema de demodulação por decisão suave com $Q=4$.
7) - Decodificação por distância de decisão suave mínima das linhas e colunas componentes do código produto, por seleção alternada adaptativa das linhas e colunas.

O código empregado é um código produto, como mostrado na figura 2.5.

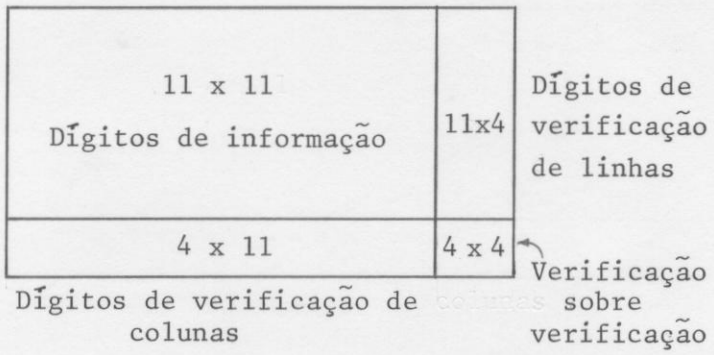


Figura 2.5 - CÓDIGO PRODUTO.

O desempenho do sistema foi muito bom em testes de ruído.

Com simulação digital de erros, uma taxa de erros de $2x10^{-1}$ nos dígitos da sequência - m foi reduzida para $4x10^{-3}$ nos dígitos de decisão na saída do correlador, e ainda reduzida para cerca de $4x10^{-7}$ depois da decodificação por decisão suave.

A figura 2.6 mostra as curvas para testes com ruído branco.

O ganho do código em $P_e = 10^{-4}$ é aproximadamente 12.3dB, enquanto que o ganho teórico máximo é cerca de 14.7dB. A diferença não é grande tendo em vista que esta degradação possibilita a realização prática.

2.2.3 - O Algorítmo de Harrison

Pode ser mostrado |40| que o processo de decisão abrupta num canal gaussiano resulta numa degradação no desempenho de aproximadamente 2dB comparado com o receptor ótimo. Segue-se que, para este canal, o máximo de melhoria a ser esperada por introduzir decisão suave no sistema, é equivalente a esta degradação. Harrison

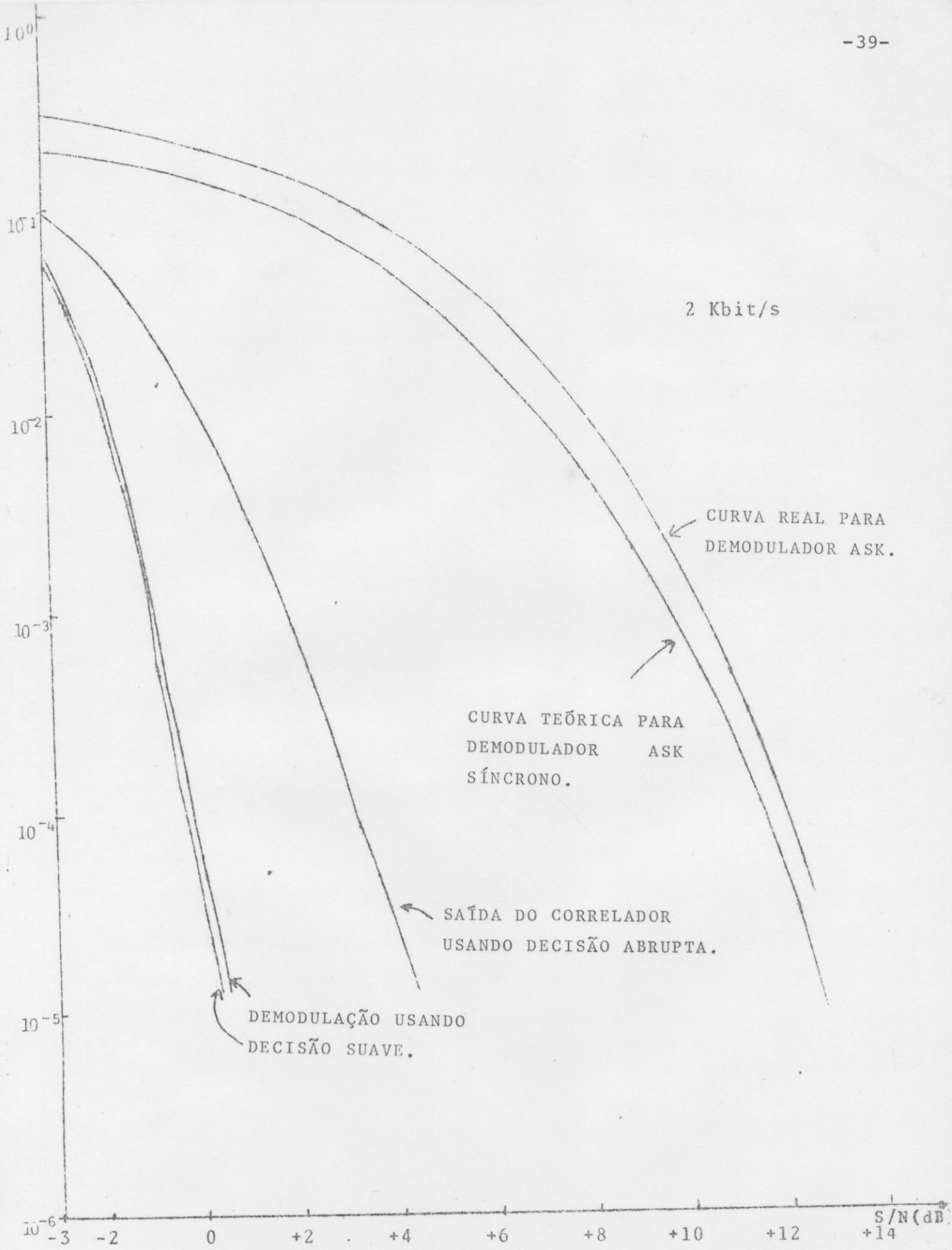


Figura 2.6 - CURVAS DE DESEMPENHO NA DECODIFICAÇÃO POR DISTÂNCIA SUAVE MÍNIMA.

|17| propôs um algorítmo de decodificação empregando decisão suave' que permite um ganho de aproximadamente 1dB, o que é um bom resultado tendo em vista a relativa simplicidade de implementação do mesmo. Neste esquema, a palavra recebida consiste da informação básica da decisão abrupta com "marcas" de confiabilidade (indicando quais as prováveis fontes de erro). Esta informação de confiabilidade disponível ao decodificador indica simplesmente quais os bits na palavra código que são os mais prováveis de terem sido afetados por erros.

Utilizou-se um código de Hamming (7,4,3) como exemplo ' para aplicação desta técnica. Desde que a complexidade do decodificador cresce rapidamente com o aumento do número de níveis de quantização, um esquema com 4 níveis é descrito.

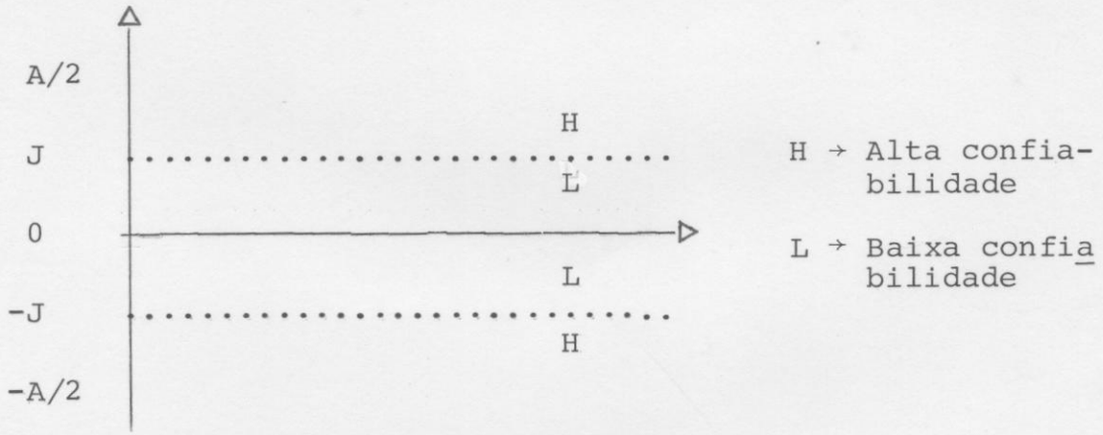


Figura 2.7 - REGIÕES DE CONFIABILIDADE.

A posição dos limiares relativos a confiabilidade é muito importante nas considerações de informação de confiabilidade. Se a tensão J é muito pequena, poucos bits recebidos serão de baixa' confiabilidade, transportando pouca informação ao decodificador. Em contrapartida, se J é uma tensão elevada, praticamente todos os bits recebidos serão de igual (baixa) confiabilidade, o que é equivalente a decisão abrupta.

Existe uma posição ótima, para cada código corretor, a qual maximiza a informação de confiabilidade entregue ao decodifica

dor. As curvas de desempenho obtidas para 3 limiares de decisão diferentes são apresentadas na figura 2.8, onde é observada a existência deste valor ótimo para J entre 40% e 50% da amplitude dos pulsos binários transmitidos.

O procedimento proposto consiste basicamente dos seguintes passos:

1) - Se a palavra recebida possui 0,1 ou mais que 2 dígitos com baixa confiabilidade, o decodificador atua desprezando a informação probabilística, i.e., empregando decisão abrupta.

2) - Se a palavra recebida tem 2 dígitos com baixa confiabilidade, então a síndrome $\underline{s}$ é calculada:

   a) Se $\underline{s} = 0$, então a palavra é suposta correta e é liberada.

   b) Se $\underline{s}$ indica um erro em uma posição de baixa confiabilidade, então a mesma é corrigida, e a palavra liberada.

   c) Se $\underline{s}$ indica um erro em uma posição de alta confiabilidade, então dois dígitos de baixa confiabilidade são invertidos e a síndrome é recalculada. Se o novo valor é $\underline{s} = 0$, é considerado que dois erros foram corrigidos, e em caso contrário, ocorreu um erro numa posição de alta confiabilidade e o decodificador deve operar normalmente.

Desta maneira o desempenho do decodificador com decisão suave deve ser pelo menos tão bom quanto o receptor equivalente empregando decisão abrupta, para todos os valores de relação sinal / ruído.

2.3 - DECODIFICAÇÃO POR DECISÃO SUAVE PARA CÓDIGOS DE BLOCO USANDO UMA TRELIÇA

Será descrito um algorítmo introduzido em 1978 por Jack Wolf |38|, o qual faz uso de decisão suave na decodificação por

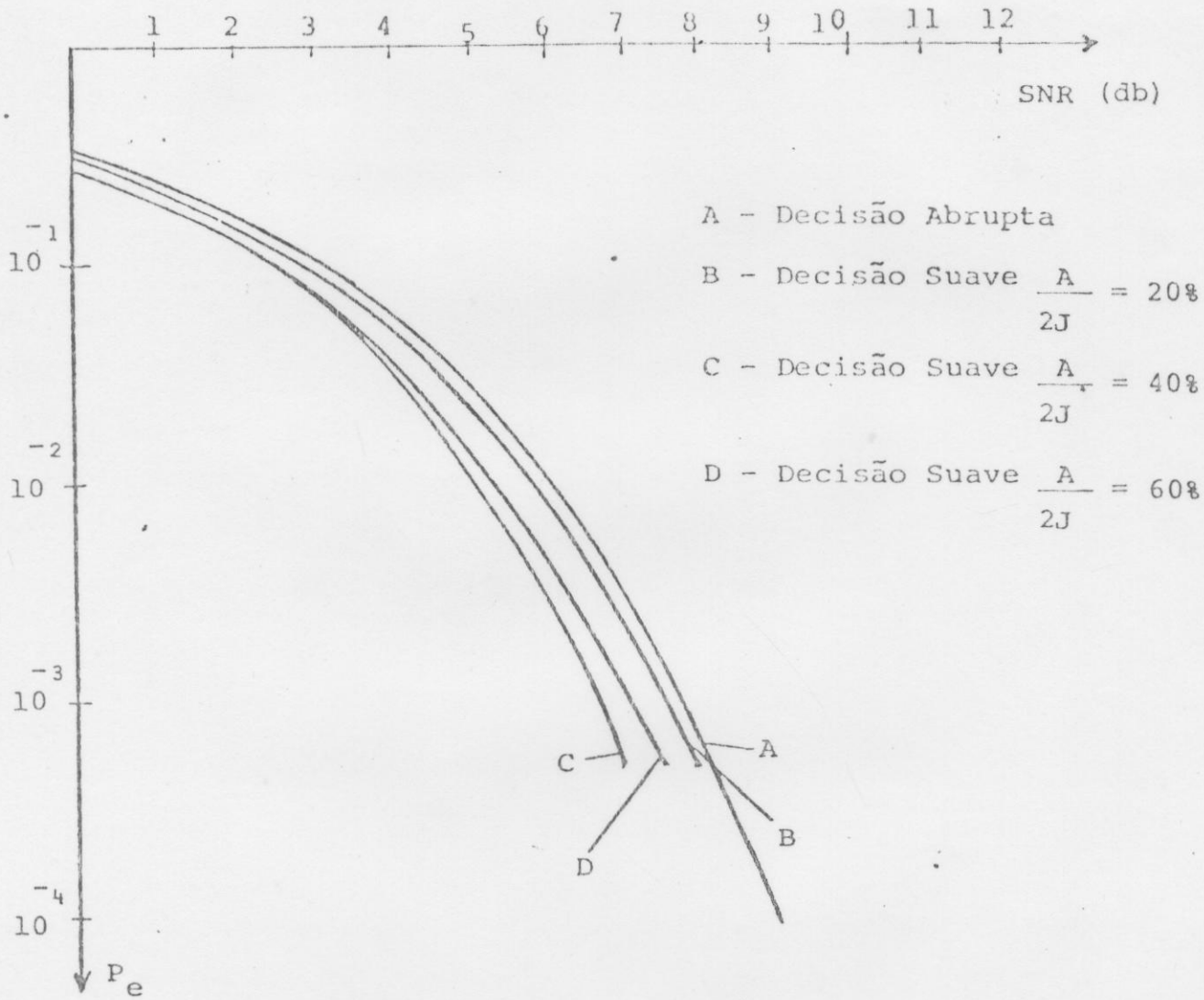


Figura 2.8 - ALGORÍTMO DE HARRISON - CURVAS DE DESEMPENHO.

máxima verossimilhança de qualquer código de bloco linear (n,k,d) sobre GF(q), e que pode ser implementado usando uma treliça tendo não mais que q^{n-k} estados.

2.3.1 - Considerações Gerais

O algorítmo apresentado faz uso da técnica de decisão suave, o que está associado ao fato de que emprega números reais (e.g., a saída analógica de filtros casados para os sinais) relacionados com cada símbolo correspondente da palavra código.

Será mostrado que a decodificação de códigos de bloco usando uma treliça é de grande interesse na decodificação de códigos com alta eficiência, visto que a complexidade da treliça é limitada superiormente por uma função do número de símbolos de paridade. Também, algumas simplificações resultam quando são considerados códigos lineares particulares, tais como códigos cíclicos ou códigos produto |38|.

2.3.2 - Treliça Associada a um Código de Bloco Linear

Seja a matriz de verificação de paridade [H] de um código de bloco linear (n,k,d) em GF(q) considerada sob a forma $[H] = |\underline{h}_1\ \underline{h}_2 \cdots \underline{h}_n|$, onde cada vetor coluna $\underline{h}_i$ possui (n-k) coordenadas em GF(q), ou seja, $|\underline{h}_i| = |h_{1i}\ h_{2i} \cdots h_{n-k\ i}|^T$ $i=1,2,\ldots,n$.

Considere um diagrama de árvore como uma coleção de nós (estados) interligados entre si por ramos unidirecionais. Cada estado a uma dada "profundidade" k é indicado por uma (n-k)-upla $\underline{S}_i(k)$ com elementos em GF(q). As linhas são traçadas da profundidade k para a profundidade k+1, considerando os nós $\{\underline{S}_i(k)\}$ a uma dada profundidade gerados como indicado no diagrama da figura 2.9.

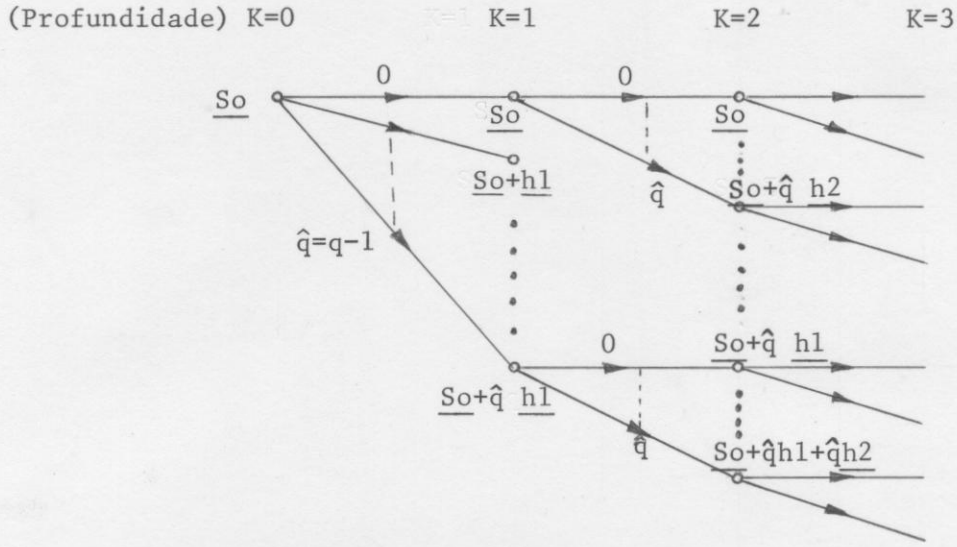


Figura 2.9 - DIAGRAMA DE ÁRVORE PARA UM CÓDIGO DE BLOCO.

Deste modo é possível verificar que a uma dada profundidade k existem os seguintes nós:

$$\underline{S}_0$$

$$\underline{S}_0 \oplus \alpha_{j\ell}\, \underline{h}_\ell \qquad 1 \leq \ell \leq k$$

$$\underline{S}_0 \oplus \alpha_{ji}\, \underline{h}_i \oplus \alpha_{j\ell}\, \underline{h}_\ell \qquad 1 \leq i < \ell \leq k$$

$$\vdots$$

etc

De uma maneira resumida, é possível escrever que a uma dada profundidade k existem os nós:

$$\underline{S}_\ell(k) = \underline{S}_0(0) \oplus \sum_{i=1}^{k} \alpha_{ji}\, \underline{h}_i, \quad \text{com} \quad \alpha_{ji} \in GF(q) \qquad (2\text{-}1)$$

Cada estado é identificado por uma (n-k)-upla em GF(q), de modo que o número de estados em qualquer profundidade não pode exceder q^{n-k}, o número de (n-k)-uplas distintas com elementos em

GF(q). Deste fato segue-se que é possível fazer uma modificação na figura 2.9, que consiste em juntar os nós indicados pela mesma (n-k)-upla, resultando em um diagrama de treliça.

Definição 2.3.2 - Uma treliça associada a um código de bloco em GF(q) é uma coleção particular de nós (ou estados) interligados por ramos unidirecionais, e agrupados em conjuntos indexados por um parâmetros k, $k = 0,1,2,\ldots,n$.

Um nó indexado por um dado valor de k é dito estar a uma profundidade k. As linhas serão traçadas entre certos pares de nós a uma profundidade k e a uma profundidade k+1, para $k = 0,1,2,\ldots,n-1$; com a direção do ramo da profundidade k para a profundidade k+1. Para qualquer profundidade k existirão no máximo q^{n-k} nós, os quais serão identificados por (n-k)-uplas, $\underline{S}_i(k)$ com elementos em GF(q) para certos valores de i. Todas as q^{n-k} (n-k)-uplas são ordenadas de 0 a $q^{n-k}-1$, com 0 referindo-se a (n-k)-upla toda zero. $\underline{S}_i(k)$ é interpretada como a i-ésima (n-k)-upla desta lista ordenada. Desde que nem todas as (n-k)-uplas podem corresponder aos nós a uma profundidade k, deve ser considerado um conjunto P_k, subconjunto dos inteiros $\{0,1,2,\ldots,q^{n-k}-1\}$, correspondendo a essas (n-k)-uplas que estão relacionadas a nós a uma profundidade k.

2.3.3 - Construção da Treliça

A treliça associada a um código possui n+1 profundidades, desde k=0 a k=n, e possui a cada profundidade q^{n-k} estados possíveis. Cada nó é caracterizado pela profundidade k e pelo estado $\underline{S}_i$ $i=0,1,2,\ldots,q^{n-k}-1$; de modo que cada nó origina q outros. Cada nó em qualquer profundidade k é obtido como combinação linear dos vetores $\underline{h}_i$ correspondentes a profundidades menores que

k, de acordo com a equação (2-1). É importante notar que $\underline{S}_i(k)$ não é univocamente definido, pois diferentes caminhos (no máximo q) podem levar a um mesmo estado.

O número de nós que são definidos a uma profundidade k é dado por q^{LI}, onde LI é o número de vetores coluna $\{h_i\}_1^k$ linearmente independentes.

A construção da treliça associada a um código de bloco é feita de acordo com o procedimento descrito a seguir:

1) - Na profundidade k=0, a treliça contém somente um nó, chamado $\underline{S}_0(0)$, a (n-k)-upla toda nula.

2) - Para cada k = 0,1,...,n-1, a coleção de nós na profundidade k+1 é obtida a partir da coleção de nós na profundidade k pela fórmula:

$$\underline{S}_\ell(k+1) = \underline{S}_i(k) \oplus \alpha_\ell \underline{h}_{k+1} \quad i\varepsilon P_k,\ \alpha_\ell \varepsilon GF(q) \qquad (2\text{-}2)$$

Para cada $i\varepsilon P_k$, as linhas de conexão são traçadas entre o nó $\underline{S}_i(k)$ e os q nós na profundidade k+1, gerados empregando a equação (2-2) acima. Cada linha é determinada pelo valor particular de α_ℓ que gerou $\underline{S}_\ell(k+1)$ a partir de $\underline{S}_i(k)$.

3) - Expurgar quaisquer nós que não tenham um caminho para o estado todo zero na profundidade k=n, e retirar todas as linhas traçadas para estes nós expurgados.

Há uma correspondência biunívoca entre cada palavra código e a sequência de α_k's, sobre qualquer caminho, do nó todo zero na profundidade 0 para o todo zero na profundidade n. Existem q^k caminhos distintos através da treliça, e cada um deles corresponde a uma palavra código. Assim, a treliça fornece um método compacto de catalogar todas as palavras código; cada palavra código distinta é representada por um caminho distinto na treliça.

<u>Exemplo 2.1</u> - O procedimento descrito é melhor ilustrado neste exemplo, onde é considerado o código binário (5,3,2) introduzido no

exemplo 1.1, cuja matriz de verificação de paridade foi determinada no exemplo 1.2 como sendo

$$[H] = \begin{vmatrix} 1 & 1 & 0 & \vdots & 1 & 0 \\ 1 & 0 & 1 & \vdots & 0 & 1 \end{vmatrix} = \left| \underline{h}_1 \ \underline{h}_2 \ \underline{h}_3 \ \underline{h}_4 \ \underline{h}_5 \right|$$

Neste caso, n=5, k=3 e q=2, de modo que $q^{n-k}=4$ e $P_k \subset \{0,1,2,3\}$.

Os nós são associados aos números inteiros na forma indicada:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \to 0 \ , \ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \to 1 \ , \ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \to 2 \ \text{ e } \ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \to 3$$

A treliça para este código é apresentada na figura 2.11.

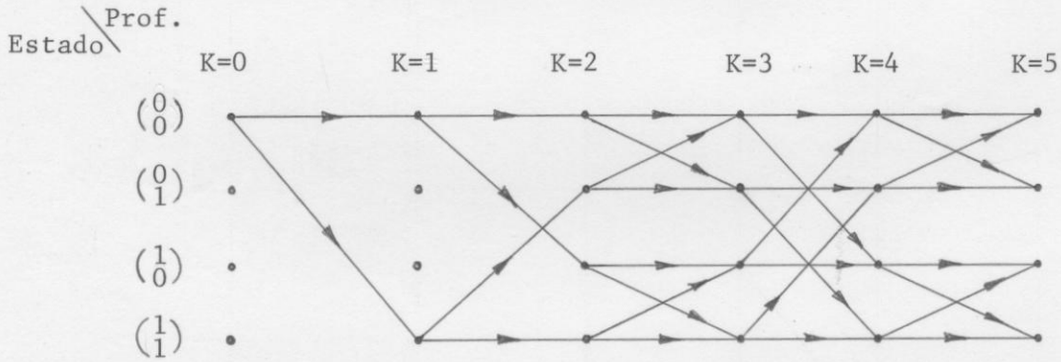


Figura 2.10 - TRELIÇA NÃO EXPURGADA PARA O CÓDIGO (5,3,2).

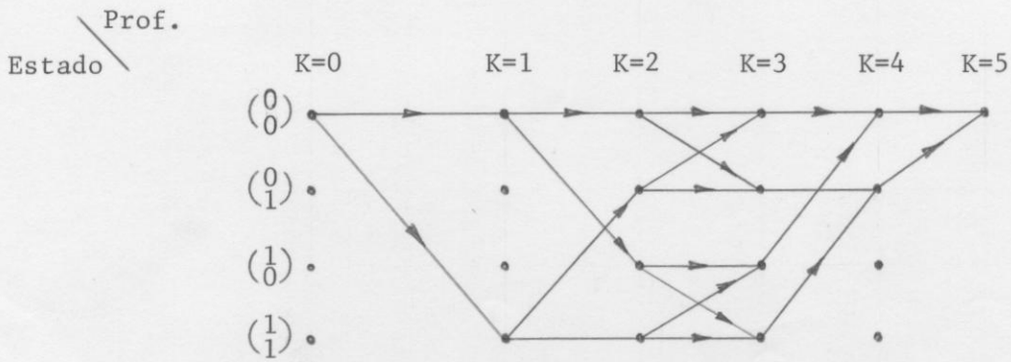


Figura 2.11 - TRELIÇA PARA O CÓDIGO DE BLOCO (5,3,2) (EXPURGADA).

A referência |25| apresenta alguns algorítmos, os quais permitem uma redução na complexidade da busca em uma treliça associada a um código de bloco. Entretanto, é evidente que deve existir um compromisso entre a complexidade e eficiência dos sistemas que empregam tais algorítmos, o qual é analisado através de simulação, na referência citada. Alguns destes procedimentos serão apresentados na seção 2.6.

2.3.4 - Decodificador de Viterbi usando a Treliça

A decodificação é executada baseada na énupla recebida $\underline{r}$, com componentes reais. É assumido que não há interferência intersimbólica, de modo que a j-ésima componente de $\underline{r}$ depende somente da j-ésima componente da palavra transmitida $\underline{c}$.

Também é assumido que a contribuição do ruído em cada uma destas componentes é descrita por variáveis aleatórias estatisticamente independentes N_i, com fdp $f_{Ni}(\ )$ $i = 1,2,\ldots,n$. Então o logarítmo da função de verossimilhança, dado a palavra código transmitida, é da forma

$$\log f_{\underline{r}|\underline{c}}(\underline{r}|\underline{c}) = \sum_{i=1}^{n} \log \left[f_{N_i}(r_i|c_i) \right] =$$

$$= \sum_{i=1}^{n} Z_i(r_i,c_i) = Z(\underline{c}) \qquad (2\text{-}3)$$

Para uma dada sequência recebida $\underline{r}$, é óbvio que uma decodificação com máxima probabilidade a posteriori é aquela que encontra a palavra código $\underline{c}$ que maximiza $Z(\underline{c})$. Seria emprego da força bruta verificar por exaustão, i.e., tentando todas as q^k possíveis palavras código.

A seguir é descrito um algorítmo recursivo pelo qual muitas palavras código podem ser descartadas no processo de decodi

ficação quando na determinação da palavra código que maximiza $Z(\underline{c})$.

A cada nó $\underline{S}_i(k)$ na profundidade k, deve ser associado um número real $V(\underline{S}_i(k))$ de acordo com as seguintes regras:

1) - Para a profundidade $k=0$, faça $V(\underline{S}_0(0)) = 0$.

2) - $\forall \ell \in P_{k+1}$, forme $V(\underline{S}_\ell(k+1))$ a partir de $V(\underline{S}_i(k))$ da seguinte maneira:

$$V(\underline{S}_\ell(k+1)) = \max_{\alpha_j \in GF(q)} \{V(\underline{S}_i(k)) + Z_{k+1}(r_{k+1}, \alpha_j)\} \qquad (2\text{-}4)$$

com $i \in P_{k,\ell}$, onde $P_{k,\ell} \subset P_k$ é um subconjunto de índices tal que, para algum α_j de $GF(q)$ tem-se

$$\underline{S}_\ell(k+1) = \underline{S}_i(k) \oplus \alpha_j \underline{h}_{k+1} \qquad (2\text{-}5)$$

3) - Retenha somente aquele caminho para o qual a fórmula (2-4) forneça o máximo.

4) - Em $k=n$, a sequência de α_j's obtida simplesmente ligando o caminho do estado todo zero na profundidade $k=0$ até o estado todo zero na profundidade $k=n$, corresponde à palavra código $\underline{c}$ que maximiza $Z(\underline{c})$.

2.4 - DECISÃO SUAVE NA DECODIFICAÇÃO QUANDO A FONTE TEM DISTRIBUIÇÃO DE PROBABILIDADE DESCONHECIDA

Será apresentado um procedimento que faz uso de coeficientes probabilísticos condicionais de confiança como medida de confiabilidade, resultando em uma regra de decodificação de máxima verossimilhança condicional.

2.4.1 - Noções Preliminares

Considere um sistema de comunicação no qual os símbolos da fonte binária tem uma distribuição de probabilidade desconhecida e são transmitidos em blocos de n dígitos através de um canal sem

memória, caracterizado por uma função de verossimilhança $f(y_i|j)$, onde $j \in \{0,1\}$, $y_i \in R$.

Uma vez a palavra binária transmitida e uma sequência $\underline{y} = (y_1,\ldots,y_n)$ recebida, o uso do método de Neymann-Pearson clássico |13| para cada y_i com um nível de confiabilidade prescrito, resulta em um dígito binário x_i. A decisão x_i não considera qualquer medida de conclusividade relativa a uma observação particular y_i; pois diferentes y_i's podem ser decodificados em um mesmo x_i, com as mesmas probabilidades de erro tipo I e tipo II, podendo cada uma das decisões possuir diferentes graus de conclusividade.

A maioria dos procedimentos de decodificação que empregam decisão suave, tratam de casos quando a distribuição da fonte é conhecida, de modo que é possível ser usada a probabilidade a posteriori como medida de confiabilidade. Contudo, é conhecido que em vários problemas práticos esse não é o caso. Wolfenson e Rocha |39| propuseram fazer uso de coeficientes de confiança condicional de Kiefer |21|, os quais têm interpretação frequentista, como uma medida de confiabilidade.

2.4.2 - Decodificação usando Confiança Condicional

O uso do lema fundamental de Neymann-Pearson permite decodificar com probabilidades de erro prescritas, cada componente da palavra recebida $\underline{y}$, resultando num vetor $\underline{x}$. Como discutido, este procedimento não expressa nenhuma conclusividade levando em conta uma observação y_i particular.

Denotando por α e β as probabilidades de erro tipo I e II, a conclusão provida pelo método é simplesmente que cada y_i deve ser decodificado em x_i com probabilidades de erro α e β. Kiefer |21| propôs um esquema de atribuir uma confiabilidade a cada decisão x_i em termos de um coeficiente de confiança condicional.

Considere uma partição Π do espaço de observações R, onde $\Pi = \{\Pi_j^s\}$ para $j \in \{0,1\}$, $s \in S$ (S um conjunto de índices).

É também definido Π^s através da relação $\Pi^s = \Pi_0^s \cup \Pi_1^s$.

Observando um dado $y_i \in R$, é procurado o elemento $\Pi_j^s \in \Pi$ no qual y_i caiu, e assim y_i é decodificado em j com confiança condicional* dada por $R_{s0} = P\{decj|\Pi^{s0},j\}$.

É assumido ainda que Π é escolhida de tal maneira que

$$P\{dec\ 0|\Pi^{s0},0\} = P\{dec\ 1|\Pi^{s0},1\} \qquad (2\text{-}6)$$

R_{s0} sendo uma probabilidade, pode ser interpretada em termos de frequência relativa. Repetindo o experimento independentemente um número muito grande de vezes de modo que se tenha um grande número de ocorrências de Π^{s0}, R_{s0} fornece aproximadamente a fração das vezes que Π^{s0} ocorreu e nas quais foi tomada uma decisão correta. Isto fornece, portanto, uma maneira objetiva de atribuir uma medida de confiabilidade a cada decisão NP. Assumindo que $\underline{y} = (y_1,\ldots,y_n)$ foi recebido e que seja verificado $y_i \in \Pi^{si}$ para cada $i = 1,2,\ldots,n$, Segue-se que cada y_i pode ser decodificado pelo procedimento convencional NP com α e β, e a medida de confiabilidade da decisão é dada por R_{si}. A questão agora é como combinar as confiabilidades R_{si}, $i = 1,2,\ldots,n$ de modo a atribuir uma medida de confiabilidade à sequência decodificada $\underline{x} = (x_1,\ldots,x_n)$. Se $\underline{x}$ é a sequência decodificada, levando em consideração que o canal é sem memória, é possível escrever

$$P\{dec\ \underline{x}\,|\,\{\Pi^{si}\}_{i=1}^{n},\ \underline{x}\} = \prod_{i=1}^{n} P\{dec\ x_i|\Pi^{si},\ x_i\}$$

$$= \prod_{i=1}^{n} R_{si} = R(\underline{x}) \qquad (2\text{-}7)$$

* Veja lista de símbolos e abreviaturas.

Desta maneira é associada uma confiabilidade dada por $R(\underline{x}) = \prod_{i=1}^{n} R_{si}$ à sequência inteira decodificada.

2.4.3 - Exemplo: Aplicação a um Canal com Ruído Branco Gaussiano

O método de decodificação apresentado é uma solução de compromisso entre minimizar a probabilidade de erro por dígito e escolher a palavra código mais provável. É possível visualizá-lo como um procedimento em dois passos, onde primeiramente o critério NP é usado, com probabilidades de erro prescritas, e então a estrutura do código é empregada juntamente com a medida de confiabilidade, de modo a encontrar a palavra código mais provável, condicionada à observação.

Nesta aplicação será considerado um canal perturbado por ruído branco gaussiano com média nula e variância unitária. A fonte bipolar transmite +1V ou -1V correspondendo aos símbolos 1 e 0, respectivamente.

Também é assumido iguais probabilidades de erros $\alpha = \beta$; a decisão NP é $y_i \underset{0}{\overset{1}{\gtrless}} 0$ com $\alpha = \beta = Q(1) = 0{,}16$.

A partição Π é escolhida de modo a permitir a máxima variabilidade nos coeficientes de confiança. No exemplo, S é escolhido como o conjunto dos possíveis valores de R_s, i.e., Π^s será indexado de modo que $R_s = s$. Portanto,

$$R_s = s = P\{\text{dec } j|\underline{y},j\} = \frac{P\{\text{dec } j,\underline{y}|j\}}{P\{\underline{y}|j\}} \qquad (2\text{-}8)$$

ou seja,

$$R_s = s = \frac{1}{1 + \dfrac{P\{\text{dec } 1-j,\underline{y}|j\}}{P\{\text{dec } j,\underline{y}|j\}}} \qquad (2\text{-}9)$$

Assim, $s = \dfrac{1}{1 + e^{\pm 2y}}$, e Π^s é escolhida como

$$\Pi^s = \{\pm\ 1/2\ \ell n\ (s^{-1}-1) \quad \text{para} \quad s > 1/2\}. \qquad (2\text{-}10)$$

Suponha, por exemplo, que $\underline{y} = (-0.2,-0.7,-0.7,0.2,1.1)$ de forma que o procedimento NP bit a bit resulta em $\underline{x}=(0,0,0,1,1)$, com coeficientes condicionais (0.6,0.8,0.8,0.6,0.9) e com confiabilidade $R(\underline{x}) = 0,207$.

Observe que devido a escolha de Π, +0.2V pode ser decodificado em 1 ou 0 com medida de confiança 0.6 ou 0.4, respectivamente. Supondo que o código (5,3,2) descrito no exemplo 1.1 é usado no contexto acima, é possível fazer uso do diagrama de treliça associado (veja exemplo 2.1, figura 2.11) para encontrar a palavra código de maior confiabilidade.

Os passos da decodificação estão esquematizados abaixo.

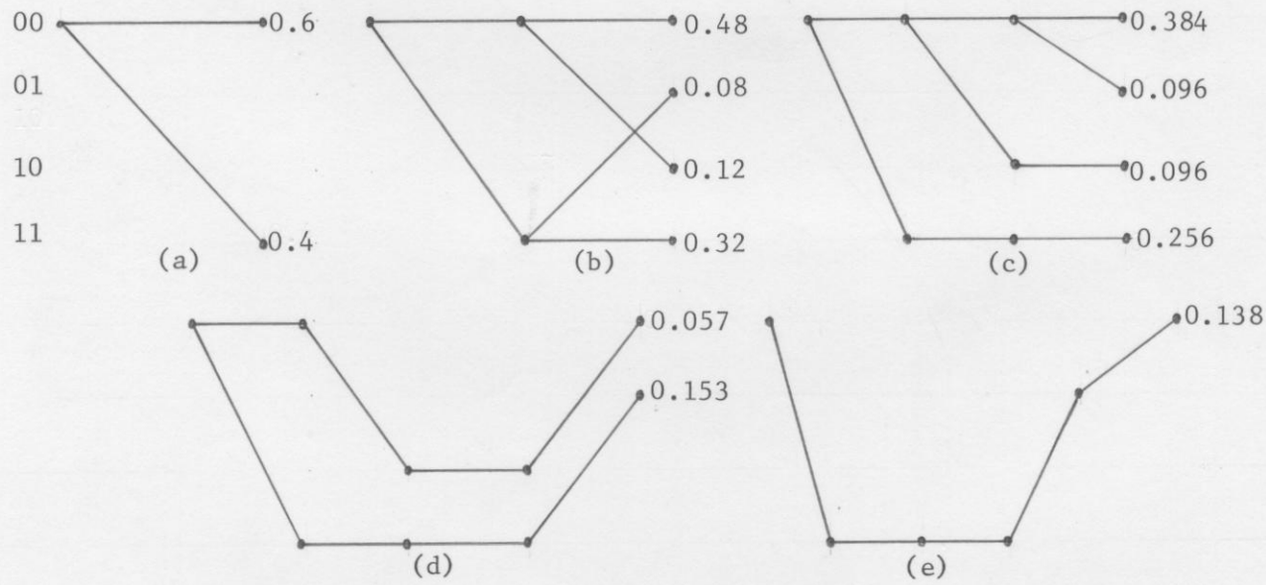


Figura 2.12 - PASSOS NA DECODIFICAÇÃO DO CÓDIGO (5,3,2) USANDO MÉTODO DE WOLFENSON-ROCHA.

Observe que um erro foi corrigido no primeiro dígito ' (R_{s1} = 0.6), pois, uma decisão abrupta estima a palavra transmitida

como (0,0,0,1,1) com $R(\underline{x}) = 0.207$, enquanto que a saída do decodificador estima (1,0,0,1,1) com $R(\underline{x}) = 0.138$. Deve também ser observado que a posição alterada foi a de menor confiabilidade, e que uma mudança na quarta posição ($R_{s4} = 0.6$) não é permitida pela estrutura do código.

2.5 - DISTÂNCIA GENERALIZADA

Será mostrado que distância generalizada constitui uma extensão da idéia de apagamento discutida anteriormente, a qual permite uma maior flexibilidade no uso da informação de confiabilidade. O emprego de distância generalizada permite que algorítmos algébricos de decodificação possam ser prontamente adaptados para fazerem uso da decisão suave. Desta forma, algumas características de decodificação probabilística são introduzidas sem sacrificar a atratividade dos procedimentos algébricos.

Serão analisados dois algorítmos para decodificar por distância mínima generalizada, sendo o segundo procedimento descrito, proposto pelo autor. Para o primeiro procedimento, introduzido por Forney |14| em 1966, foram também desenvolvidas cotas exponenciais para a probabilidade de não decodificar corretamente.

2.5.1 - Medidas de Distância

Os algorítmos de decodificação são formulados, geralmente, em termos da escolha da palavra-código mais próxima da palavra recebida, onde a "proximidade" é definida por alguma medida de distância. Duas importantes medidas de distância são apresentadas nesta seção. A distância de Hamming, já introduzida na definição 1.2.2b, será reapresentada de uma maneira mais formal.

Definição 2.5.1a - A distância de Hamming $D_H(\underline{f},\underline{g})$ entre duas palavras $\underline{f}$ e $\underline{g}$ é simplesmente o número de posições em que elas diferem; i.e., $D_H(\underline{f},\underline{g}) = \sum_{i=1}^{n} d_H(f_i,g_i)$

Onde

$$d_H(f_i,g_i) = \begin{cases} 0 & \text{se} \quad f_i = g_i \\ 1 & \text{se} \quad f_i \neq g_i \end{cases}$$

Suponha que o receptor decide, para cada amostra recebida, qual das q letras foi a mais provavelmente enviada, uma escolha que corresponde a um elemento do GF(q) rotulado por r_i. Esta escolha é então associada a uma classe de confiabilidade c_j de acordo com a certeza que se tem da mesma ter sido correta. A cada classe c_j são associados dois parâmetros β_{cj} e β_{ej} tais que $0 \leq \beta_{cj} \leq \beta_{ej} \leq 1$. Contudo, será mostrado que somente a quantidade $\alpha_j = \beta_{ej} - \beta_{cj}$, chamada de peso da classe c_j, é significante. Vê-se claramente que $(\forall j)$ é verificada a relação $0 \leq \alpha_j \leq 1$.

Definição 2.5.1b - Distância generalizada $D_G(\underline{r},\underline{f})$ entre a palavra recebida $\underline{r}$ e a palavra código $\underline{f}$ é definida como sendo

$$D_G(\underline{r},\underline{f}) \triangleq \sum_{i=1}^{n} d_G(r_i,f_i)$$

Onde

$$d_G(r_i,f_i) = \begin{cases} \beta_{cj} & \text{se} \quad r_i = f_i \ , \ r_i \in c_j \\ \beta_{ej} & \text{se} \quad r_i \neq f_i \ , \ r_i \in c_j \end{cases}$$

Deve ser observado que esta distância não constitui uma métrica |2|, enquanto que D_H representa uma métrica.

Em geral, as classes para as quais $\alpha=1$ serão referidas como classes de plena confiança, e aquelas para as quais $\alpha=0$ como classes não confiáveis. Valores intermediários de α_j correspondem a níveis intermediários de confiabilidade. Também serão ordenados os números das classes em ordem decrescente com relação ao seu peso, de

maneira que se $j < k$, então $\alpha_j \geq \alpha_k$.

2.5.2 - Decodificação por Distância Mínima Generalizada I

Algumas das propriedades da distância generalizada serão discutidas a seguir, as quais permitem o estabelecimento de um decodificador usando distância mínima generalizada. Cotas exponenciais de erros para o algorítmo estabelecido serão também desenvolvidas.

2.5.2.1 - Propriedades da Distância Generalizada

Inicialmente suponha que durante a transmissão de uma palavra $\underline{f}$ de um código de bloco (n,k,d), o número de letras recebidas corretamente $(r_i = f_i)$ e postas na classe c_j é n_{cj}, e o número das recebidas incorretamente $(r_i \neq f_i)$ e postas nesta classe c_j é n_{ej}.

Teorema 2.1 - Se $\underline{f}$ é enviada e n_{cj} e n_{ej} são tais que se verifica $\sum_j \left\{(1-\alpha_j)n_{cj} + (1+\alpha_j)n_{ej}\right\} < d$, então $D_G(\underline{r},\underline{f}) < D_G(\underline{r},\underline{g})$ para todas palavras código $\underline{g} \neq \underline{f}$.

Prova - Pela definição de distância generalizada segue-se que

$$D_G(\underline{r},\underline{f}) = \sum_j \left\{\beta_{cj}\, n_{cj} + \beta_{ej}\, n_{ej}\right\} \tag{2-11}$$

Para qualquer palavra código $\underline{g} \neq \underline{f}$ serão considerados três conjuntos de índices, de acordo com

$$i \in \begin{cases} s_0 & \text{se} \quad f_i = g_i \\ s_{cj} & \text{se} \quad f_i \neq g_i,\ r_i = f_i \quad \text{com} \quad r_i \in c_j \\ s_{ej} & \text{se} \quad f_i \neq g_i,\ r_i \neq f_i \quad \text{com} \quad r_i \in c_j \end{cases}$$

O número de letras $|s_{cj}|$ e $|s_{ej}|$ dos conjuntos s_{cj} e s_{ej} satisfazem as relações $|s_{cj}| \leq n_{cj}$ e $|s_{ej}| \leq n_{ej}$. Então:

$$d_G(r_i,g_i) \geq 0 = d_H(g_i,f_i) \qquad i \in s_0 \qquad (2\text{-}12)$$

$$d_G(r_i,g_i) = \beta_{ej} = d_H(g_i,f_i)-1 + \beta_{ej} \qquad i \in s_{cj} \qquad (2\text{-}13)$$

$$d_G(r_i,g_i) \geq \beta_{cj} = d_H(g_i,f_i)-1 + \beta_{cj} \qquad i \in s_{ej} \qquad (2\text{-}14)$$

Operando com as equações (2-12) a (2-14) acima, obtém - se:

$$D_G(\underline{r},\underline{g}) \geq D_H(\underline{f},\underline{g}) - \sum_j \left\{(1-\beta_{ej})|s_{cj}| + (1-\beta_{cj})|s_{ej}|\right\}$$

$$\geq d - \sum_j \left\{(1-\beta_{ej})n_{cj} + (1-\beta_{cj})n_{ej}\right\} \qquad (2\text{-}15)$$

Utilizando a hipótese $d > \sum_j (1-\beta_{ej}+\beta_{cj})n_{cj}+(1-\beta_{cj}+\beta_{ej})n_{ej}$ para enfraquecer a desigualdade (2-15), tem-se

$$D_G(\underline{r},\underline{g}) > \sum_j \left\{n_{cj}\ \beta_{cj} + n_{ej}\ \beta_{ej}\right\} = D_G(\underline{r},\underline{f})$$

C.Q.D

É óbvio que se o critério de distância mínima generalizada for utilizado como critério de decodificação, não serão cometidos erros nos casos onde n_{cj} e n_{ej} são tais que a desigualdade do teorema 2.1 é satisfeita.

No caso especial em que há somente uma classe $\alpha_1 = 1$, n_{e1} é o número total t de erros em $\underline{r}$, de maneira que o teorema' torna-se $D_G(\underline{r},\underline{f}) < D_G(\underline{r},\underline{g})\ \forall \underline{g} \neq \underline{f}$ se $2t < d$. Se for permitida a inclusão de uma classe de apagamentos $\alpha_2 = 0$, cujo número total é $s = n_{c2} + n_{e2}$, o teorema pode ser reescrito como

$$D_G(\underline{r},\underline{f}) < D_G(\underline{r},\underline{g})\ \forall \underline{g} \neq \underline{f} \quad \text{se} \quad 2t + s < d.$$

Estas são condições familiares que asseguram a não ocorrência de erros quando for utilizada a decodificação por distância mínima.

Definição 2.5.2a - É dito que a palavra código está "dentro da distância mínima da palavra recebida" quando a desigualdade do teorema 2.1 é aparentemente satisfeita.

O teorema mostra que há no máximo uma palavra código dentro da distância mínima de qualquer palavra recebida. Assim, se o decodificador encontra alguma palavra código dentro da distância mínima da palavra recebida, ele pode imediatamente anunciar que esta é a palavra decodificada.

2.5.2.2 - O Algorítmo de Forney

Forney Jr.|14| propôs um algorítmo prático que usa distância mínima generalizada como critério de decodificação, e que funciona se há uma palavra dentro da distância mínima da palavra recebida. Aqui é assumida a existência de algorítmos de decodificação que são capazes de manipular com apagamentos e erros, e que operam se o número s de apagamentos e o número t de erros são tais que $2t + s < d$.

Suponha, para este propósito, que as letras recebidas em algumas classes são consideradas como apagamentos, e as restantes como completamente confiáveis. Denotando por

$$E = \{j | c_j \text{ é não confiável}\}, \text{ e } R = E^c = \{j | c_j \text{ é de plena confiança}\}$$

é tentado decodificar a palavra recebida pelo algorítmo de erros-e-apagamentos, o qual estará apto para decodificar se

$$2 \sum_{j\varepsilon R} n_{ej} + \sum_{j\varepsilon E} (n_{cj} + n_{ej}) < d. \qquad (2\text{-}16)$$

Se isto acontece, verifica-se, então, se a palavra código obtida está dentro da distância mínima da palavra recebida. Caso ela esteja, então foi determinada a única palavra dentro da distância mínima.

Não há uma única atribuição provisória de apagamentos para a qual este método tem sucesso. Contudo, o teorema a seguir e seu corolário mostram que um pequeno número de tais tentativas apresentam êxito em encontrar a única palavra código dentro da distância mínima, se existe uma. Como anteriormente, as classes são consideradas ordenadas de acordo com ordem decrescente de confiabilidade.

Seja J o número total de classes, e um vetor de J dimensões $\underline{\alpha}$ definido como $\underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_J)$. São também definidos os conjuntos $R_b = \{j | j \leq b\}$ e $E_b = \{j | j \geq b+1\}$ com $0 \leq b \leq J$.

É considerado que $\underline{\alpha}_b$, um vetor J-dimensional com 1's nas primeiras b posições e zeros nas demais, representa uma atribuição provisória correspondente a $R = R_b$ e $E = E_b$. A idéia do teorema 2.2 abaixo é que $\underline{\alpha}$ está dentro de uma envoltória convexa, cujos pontos extremos são $\underline{\alpha}_b$'s, enquanto a expressão $f(\underline{\alpha})$ a ser definida é uma função linear de $\underline{\alpha}$, que deve assumir valores mínimos em algum ponto extremo, i.e., para uma das atribuições provisórias $\underline{\alpha}_b$.

<u>Teorema 2.2</u> - Se $\sum_{j=1}^{J} \left\{(1-\alpha_j)n_{cj} + (1+\alpha_j)n_{ej}\right\} < d$ e $\alpha_j \geq \alpha_k$ para $j < k$, existe algum inteiro b tal que a desigualdade $2 \sum_{j=1}^{b} n_{ej} + \sum_{j=b+1}^{J} (n_{cj} + n_{ej}) < d$ é satisfeita.

<u>Prova</u> - Inicialmente é definida uma função do vetor $\underline{\alpha}$,

$$f(\underline{\alpha}) \triangleq \sum_{j=1}^{J} \left\{(1-\alpha_j)n_{cj} + (1+\alpha_j)n_{ej}\right\} < d \qquad (2\text{-}17)$$

a qual satisfaz as seguintes propriedades:

i - f é uma função linear do vetor J-dimensional $\underline{\alpha}$

ii- note que $f(\underline{\alpha}_b) = 2 \sum_{j=1}^{b} n_{ej} + \sum_{j=b+1}^{J} (n_{cj} + n_{ej})$.

O teorema é provado supondo que $(\forall b)\ f(\underline{\alpha}_b) \geq d$, $0 \leq b \leq J$, e exibindo uma contradição.

Para isto, sejam
$$\begin{cases} \lambda_0 = 1 - \alpha_1 \\ \lambda_b = \alpha_b - \alpha_{b+1}, \quad 1 \leq b \leq J-1 \\ \lambda_J = \alpha_J \end{cases}$$

É facilmente verificado que $0 \leq \lambda_b \leq 1$ para $b = 0,1,\ldots,J$ e que $\sum_{b=0}^{J} \lambda_b = 1$.

Ora, $\underline{\alpha} = \sum_{b=0}^{J} \lambda_b \underline{\alpha}_b$, de modo que é possível escrever

$$f(\underline{\alpha}) = f(\sum_{b=0}^{J} \lambda_b \underline{\alpha}_b) = \sum_{b=0}^{J} \lambda_b f(\underline{\alpha}_b) \geq d \sum_{b=0}^{J} \lambda_b = d$$

CONTRADIÇÃO !

Portanto, conclui-se que $f(\underline{\alpha}_b) < d$ para pelo menos um valor de b, $0 \leq b \leq J$. C.Q.D.

Este teorema prova que se há alguma palavra código dentro da distância mínima da palavra recebida, então deve haver alguma atribuição provisória, a qual permite que o decodificador de erros-e-apagamentos tenha sucesso em decodificar de acordo com o critério de distância mínima generalizada. Mas um decodificador de erros-e-apagamentos tem êxito se e somente se existem aparentemente:

Nenhum erro	e	$d - 1$	apagamentos
Um erro	e	$d - 3$	apagamentos
.....		.....	
t_0 erros	e	$d - 2t_0 - 1$	apagamentos, onde $t_0 = \left\lfloor \frac{d-1}{2} \right\rfloor$.

<u>Corolário</u> - Bastam $t_0 + 1 \leq \frac{d+1}{2}$ tentativas de possíveis atribuições provisórias para que se tenha sucesso em decodificar qualquer palavra recebida que está dentro da distância mínima pelo critério

de distância mínima generalizada, não importando quantas classes de confiabilidade existam.

Segue-se, portanto, que o número máximo de tais processos é somente proporcional a d. Ademais, muitos dos processos (talvez todos) podem ter êxito, de modo que o número médio de processos pode ser apreciavelmente menor que o máximo.

2.5.2.3 - Cotas (exponenciais) de Erros para o Algorítmo de Forney

Para o algorítmo proposto por Forney |14| existem desenvolvidas cotas "apertadas" para a probabilidade de não decodificar corretamente, onde "não decodificar corretamente" significa decodificação incorreta ou incapacidade de decodificar. Este último evento ocorre quando não há palavra código dentro da distância mínima da palavra recebida.

Considere uma variável aleatória y_i, a qual para cada letra transmitida assume valores:

$$y_i = \begin{cases} 1-\alpha_j & \text{se a letra é recebida corretamente e posta em } c_j \\ 1+\alpha_j & \text{se a letra é recebida incorretamente e posta em } c_j \end{cases}$$

Assumindo um canal sem memória, estas variáveis aleatórias $\{y_i\}_1^n$ resultam iid.

Sejam $p_{cj}=pr\{y_i=(1-\alpha_j)\}$ e $p_{ej}=pr\{y_i=(1+\alpha_j)\}$ (2-18)

A função geradora de momentos |8| g(s) para a variável aleatória y_i é dada por

$$g(s) = E\{e^{s_{yi}}\} = \sum_j p_{cj}\, e^{s(1-\alpha_j)} + p_{ej}\, e^{s(1+\alpha_j)} \qquad (2\text{-}19)$$

e a função geradora semi-invariante |8| correspondente, expressa por $\mu(s) = \ell n g(s)$.

Já é sabido que não haverá erros na decodificação se n_{cj} e n_{ej} são tais que a desigualdade do teorema 2.1 é satisfeita, o que é o mesmo que requerer que $\sum_{i=1}^{n} y_i < d$. Deste fato segue-se que

$$pr(ndc) = pr\{\sum_{i=1}^{n} y_i \geq d\} \qquad (2\text{-}20).$$

A cota de Chernoff |8| é uma cota exponencialmente apertada para a probabilidade de que a soma de variáveis aleatórias iid exceda um certo número; no caso, a cota resulta em

$$pr(ndc) \leq \exp - n\,|s\delta - \mu(s)| \qquad (2\text{-}21)$$

válida $\forall s \geq 0$, onde $\delta = d/n$.

É possível tornar a cota mais apertada maximizando o expoente pela escolha de s e dos α_j's:

$$E(\delta) = \max_{s,\alpha_j} |s\delta - \mu(s)| \qquad (2\text{-}22)$$

Primeiro, a maximização será feita sobre α_j; e desde que $\mu(s) = \ell n g(s)$, isto é realizado minimizando $g(s)$.

$$\frac{\partial}{\partial \alpha_j} g(s) = s\, p_{cj} e^{s(1-\alpha_j)} + s\, p_{cj} e^{s(1+\alpha_j)} = 0 \qquad (2\text{-}23)$$

Os α_j's ótimos são então determinados a partir de (2-23) acima, lembrando da restrição $0 \leq \alpha_j \leq 1$, o que resulta em

$$\alpha_j = \begin{cases} 0 & \text{se} \quad L_j \leq 0 \\ L_j/2s & \text{se} \quad 0 \leq L_j \leq 2s \\ 1 & \text{se} \quad L_j \geq 2s \end{cases} \qquad (2\text{-}24)$$

onde $$L_j = \ell n \frac{p_{cj}}{p_{ej}} = \ell n \frac{pr\{r_i \text{ correta} | r_i \varepsilon c_j\}}{pr\{r_i \text{incorreta} | r_i \varepsilon c_j\}}$$

Assim, a atribuição ótima dos pesos envolve apagamento' de qualquer recepção para a qual a probabilidade da escolha correta p seja menor que 1/2 ; considera plenamente confiável qualquer recepção para a qual p seja maior que um limiar (que depende de S), e para valores intermediários assume um valor proporcional ao ℓog da razão de verossimilhança $\ell n\ p/(1-p)$. Portanto, exceto pelas limitações nos extremos, a decodificação por distância mínima generalizada é um método que utiliza o ℓog da razão de verossimilhança em esquemas de decodificação algébrica.

Suponha definidas as classes $\begin{cases} j\varepsilon R & \text{se } L_j > 2s \\ j\varepsilon E & \text{se } L_j \leq 0 \\ j\varepsilon G & \text{noutros casos} \end{cases}$

Segue-se, então, que

$$g_{opt}(s) = e^{2s} p_e(s) + e^s p_x(s) + e^s p_g(s) + p_c(s) \tag{2-25}$$

onde,

$$p_e(s) = \sum_{j\varepsilon R} p_{ej} \quad , \quad p_c(s) = \sum_{j\varepsilon R} p_{cj} \tag{2-26}$$

$$p_x(s) = \sum_{j\varepsilon E} (p_{ej} + p_{cj}) \tag{2-27}$$

$$p_g(s) = \sum_{j\varepsilon G} 2\sqrt{p_{ej}\ p_{cj}} \tag{2-28}$$

Finalmente, assumindo $\mu_{opt}(s) = \ell n\ g_{opt}(s)$ e tendo em vista a equação (2-22), tem-se que

$$E(\delta) = \max_s |s\delta - \mu_{opt}(s)| \tag{2-29}$$

Agora otimizando pela escolha de s, observa-se que

para o s ótimo verifica-se a relação

$$\delta = \mu'_{opt}(s) = \frac{g'_{opt}(s)}{g_{opt}(s)} \qquad (2\text{-}30)$$

Como $\mu'_{opt}(s)$ é monotonicamente crescente com s, pois, $\mu(s)$ é convexa | 8 |, a equação (2-30) acima admite solução com $s \geq 0$ se e somente se $\delta > \mu'_{opt}(0)$. É concluido, portanto, que a menor distância mínima (de Hamming) para a qual a decodificação por distância mínima generalizada pode operar é dada por $n\mu'_{opt}(0)$.

2.5.3 - Decodificação por Distância Mínima Generalizada II

Neste seção são feitas algumas considerações adicionais sobre distância generalizada, apresentando um algorítmo de decodificação por distância mínima para códigos de bloco, o qual faz uso da treliça associada ao código.

O algorítmo aqui apresentado estabelece um procedimento equivalente ao proposto na seção anterior, no sentido de que também escolhe a palavra código mais próxima da palavra recebida, quando é utilizada como medida de proximidade a distância generalizada. Deve ser mencionado, entretanto, que existem algumas dificuldades práticas na implementação do decodificador descrito pelo teorema 2.2 da seção anterior. As $t_0 + 1$ tentativas de decodificação do corolário deste teorema podem resultar em até $t_0 + 1$ palavras código diferentes. A classe de algorítmos apresentados por Forney sugere o uso de técnicas algébricas para gerar um certo número de palavras código próximas em algum sentido da palavra recebida. O problema crucial é encontrar uma técnica eficiente de gerar o conjunto de palavras código, o qual tenha alta probabilidade de conter a palavra código mais provável, dado a palavra recebida. Será mostrado que o procedimento descrito a seguir se apresenta mais simples em termos de com-

plexidade do decodificador.

2.5.3.1 - Considerações Introdutórias

Em princípio, uma mudança na notação será realizada por conveniência, de modo a torná-la adequada ao problema. Considere R o espaço de observações e uma partição de R denotada por $\{\Pi_j^s\}$, $j \in \{0,1\}$ e $s \in S$, onde S é um conjunto de índices. $\Pi^s = \Pi_0^s \cup \Pi_1^s$ denota uma classe de confiabilidade, a qual são associados dois parâmetros β_{e_s} e β_{c_s}, $0 \leq \beta_{c_s} \leq \beta_{e_s} \leq 1$; e a quantidade $\alpha_s = \beta_{e_s} - \beta_{c_s}$ é dita ser o peso da classe Π^s. Também será considerado um vetor $\underline{\alpha}^{(\ )} = (\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(n)})$ definido pelo peso das classes nas quais as componentes da palavra recebida $\underline{r} = (r_1, r_2, \ldots, r_n)$ se encontram; ou seja, se $r_i \in \Pi^{s_0}$ então $\alpha^{(i)} = \alpha_{s_0}$. Assim, a notação (i) indica que se faz referência a classe de confiabilidade a qual r_i pertence.

Uma das perguntas importantes que surgem quando é dada uma partição $\{\Pi_j^s\}$ do espaço de observação R, é como atribuir valores aos parâmetros β_{e_s} e β_{c_s} para cada classe de confiabilidade Π^s. É claro que o emprego da teoria da decisão seria oportuno aqui, usando apropriadamente o conceito de função utilidade |13|. Contudo, será proposta uma maneira de se atribuir valores para os pesos α_s de cada classe Π^s da partição, de forma a representarem um grau de confiabilidade associado à classe.

2.5.3.2 - Sobre Atribuição de Pesos a Classes de Confiabilidade

Dada uma partição qualquer $\{\Pi^s\}$, é proposto associar a cada elemento Π^s da partição os seguintes parâmetros:

classe de confiabilidade Π^s =>

$\beta_{e_s} = p\{\text{dec correto}|r_i \varepsilon \Pi^s\}$ e $\beta_{c_s} = p\{\text{dec com erro}|r_i \varepsilon \Pi^s\}$.

Deve ser observado que $0 \leq \beta_{c_s} \leq \beta_{e_s} \leq 1$, e que esta atribuição implica em $\beta_{e_s} + \beta_{c_s} = 1$.

De acordo com a definição 2.5.1b, a distância generalizada de Forney entre a palavra recebida $\underline{r}$ e uma palavra código $\underline{f}$ será expressa por

$$D_G(\underline{r},\underline{f}) = \sum_{i=1}^{n} d_G(r_i,f_i) \tag{2-31}$$

onde,

$$d_G(r_i,f_i) = \begin{cases} \beta_{e_s} = p\{\text{dec correto}|r_i \varepsilon \Pi^s\} & r_i \neq f_i, \quad r_i \varepsilon \Pi^s \\ \beta_{c_s} = p\{\text{dec com erro}|r_i \varepsilon \Pi^s\} & r_i \neq f_i, \quad r_i \varepsilon \Pi^s \end{cases} \tag{2-32}$$

Inicialmente será analisado o caso binário, onde é assumido que a partição é simétrica, ou seja, $\{\Pi^s\}$ é escolhida de tal maneira que

$$p\{\text{dec } 0|\Pi^s,0\} = p\{\text{dec } 1|\Pi^s,1\} \tag{2-33}$$

Para uma classe Π^{s0}, o coeficiente de confiança condicional de Kiefer |21| é dado por $R_{s0} = p\{\text{dec } j|\Pi^{s0},j\}$.

Neste caso particular, (2-32) pode ser reescrita na forma

$$d_G(r_i,f_i) = \begin{cases} \beta_{e_s} = p\{\text{dec } j|\Pi^{s0},j\} & r_i \neq f_i, \; r_i \varepsilon \Pi^{s0} \\ \beta_{c_s} = 1-p\{\text{dec } j|\Pi^{s0},j\} & r_i = f_i, \; r_i \varepsilon \Pi^{s0} \end{cases} \tag{2-34}$$

Então para a classe Π^{s0}, o peso associado α_{s0} é dado por $\alpha_{s0} = 2.p\{\text{dec } j|\Pi^{s0},j\} - 1 = 2\,R_{s0} - 1$ (2-35)

Notando que $0.5 \leq R_{s0} \leq 1$, tem-se que $0 \leq \alpha_{s0} \leq 1$.

Por outro lado,

$$R_{s0} = p\{dec\ j|\Pi^{s0},j\} = \frac{p\{dec\ j,\Pi^{s0}|j\}}{p\{\Pi^{s0}|j\}} \qquad (2\text{-}36)$$

ou seja,

$$p\{dec\ j|\Pi^{s0},j\} = \frac{p\{dec\ j,\Pi^{s0}|j\}}{p\{dec\ j,\Pi^{s0}|j\} + p\{dec\ 1-j,\Pi^{s0}|j\}} \qquad (2\text{-}37)$$

Desta forma, deve ser assumido que

$$\beta_{e_s} = \frac{1}{1+\dfrac{p\{dec\ 1-j,\Pi^{s0}|j\}}{p\{dec\ j,\Pi^{s0}|j\}}} = \frac{1}{1+\dfrac{p\{\Pi^{s0}_{i-j}|j\}}{p\{\Pi^{s0}_{j}|j\}}} \qquad (2\text{-}38)$$

Definindo $\emptyset_{\ell|j}$, a razão de verossimilhança na classe Π^{s0},

$$\emptyset_{\ell|j} \overset{\Delta}{=} \frac{p\{\Pi^{s0}_{\ell}|j\}}{p\{\Pi^{s0}_{j}|j\}} \qquad (2\text{-}39)$$

e observando que para partição simétrica $\emptyset_{0|1} = \emptyset_{1|0} = \emptyset$, a equação (2-38) pode ser escrita como

$$\beta_{e_s} = \frac{1}{1-\emptyset}\ ;\ \text{consequentemente,}\ \beta_{c_s} = 1 - \frac{1}{1-\emptyset} = \frac{\emptyset}{1-\emptyset}$$

de modo que o peso associado a classe Π^{s0} será dado por

$$\alpha_{s0} = \beta_{e_s} - \beta_{c_s} = \frac{1-\emptyset}{1+\emptyset} = \rho \qquad (2\text{-}40)$$

É interessante observar que o algorítmo de Hartmann - Rudolph, descrito no capítulo III, utiliza esta quantidade ρ no processo de decodificação.

A generalização deste procedimento para o caso multiní

vel e para casos onde a partição do espaço de observações não é simétrica será realizada a seguir. Aqui são supostas conhecidas as probabilidades a priori dos símbolos da fonte, denotadas por $\{p_j\}_0^{q-1}$.

A equação (2-39) resulta em

$$d_G(r_i,f_i)=\begin{cases}\beta_{e_s}=\sum_{j=0}^{q-1} p_j\ p\{\text{dec } j|\Pi^s,j\} & r_i\neq f_i,\ r_i\varepsilon\Pi^s\\ \beta_{c_s}=1-\sum_{j=0}^{q-1} p_j\ p\{\text{dec } j|\Pi^s,j\} & r_i=f_i,\ r_i\varepsilon\Pi^s\end{cases} \tag{2-41}$$

e o peso da classe Π^{s0} será agora determinado de acordo com

$$\alpha_{s0} = 2.\sum_{j=0}^{q-1} p_j\ p\{\text{dec } j|\Pi^{s0},j\} - 1 \tag{2-42}$$

$$= \sum_{j=0}^{q-1} p_j\ 2\ p\{\text{dec } j|\Pi^{s0},j\} - \sum_{j=0}^{q-1} p_j \tag{2-43}$$

$$= \sum_{j=0}^{q-1} p_j\,[2\ p\{\text{dec } j|\Pi^{s0},j\}-1] \tag{2-44}$$

No caso multinível, a equação correspondente a (2-36) é

$$p\{\text{dec } j|\Pi^{s0},j\} = \frac{p\{\Pi_j^{s0}|j\}}{\sum_{\ell=0}^{q-1} p\{\Pi_\ell^{s0}|j\}} \tag{2-45}$$

$$= \frac{1}{\sum_{\ell=0}^{q-1} \frac{p\{\Pi_\ell^{s0}|j\}}{p\{\Pi_j^{s0}|j\}}} \tag{2-46}$$

Relembrando da definição de $\emptyset_{\ell|j}$ dada pela equação (2-39), tem-se que

$$p\{\text{dec } j|\Pi^{s0},j\}= \frac{1}{\sum_{\ell=0}^{q-1} \emptyset_{\ell|j}} \tag{2-47}$$

Desta forma, o termo entre colchetes na equação (2-44)' é

$$2.p\{\text{dec } j|\Pi^{s0},j\}-1 = 2.\frac{1}{\sum_{\ell=0}^{q-1} \phi_{\ell|j}} - 1 \qquad (2\text{-}48)$$

$$= \frac{2 - \sum_{\ell=0}^{q-1} \phi_{\ell|j}}{\sum_{\ell=0}^{q-1} \phi_{\ell|j}} \qquad (2\text{-}49)$$

usando o fato que $\phi_{j|j} = 1$, (2-49) pode ser dada por

$$2.p\{\text{dec } j|\Pi^{s0},j\} - 1 = \frac{1 - \sum_{\ell=0}^{q-1} \phi_{\ell|j}}{1 + \sum_{\ell=0}^{q-1} \phi_{\ell|j}}, \ \ell\neq j \qquad (2\text{-}50)$$

Esta expressão será definida como ρ_j multinível,

$$\rho_j \triangleq \frac{1 - \sum_{\substack{\ell=0\\ \ell\neq j}}^{q-1} \phi_{\ell|j}}{1 + \sum_{\substack{\ell=0\\ \ell\neq j}}^{q-1} \phi_{\ell|j}} \qquad (2\text{-}51)$$

Então, de acordo com (2-44) o peso da classe Π^{s0} será' calculado pela fórmula

$$\underline{\alpha_{s0} = \sum_{j=0}^{q-1} p_j \rho_j} \qquad (4\text{-}52)$$

No caso binário, $\rho_0 = \dfrac{1 - \phi_{1|0}}{1 + \phi_{1|0}}$ e $\rho_1 = \dfrac{1 - \phi_{0|1}}{1 + \phi_{0|1}}$,

de modo que quando a partição é não simétrica, tem-se $\rho_0 \neq \rho_1$,

como exemplificado na figura 2.13 a título de ilustração. No caso onde (2-33) se verifica, $\emptyset_{0|1} = \emptyset_{1|0} = \emptyset$ e consequentemente $\rho_0 = \rho_1 = \rho$; portanto, (2-52) se reduz a (2-40).

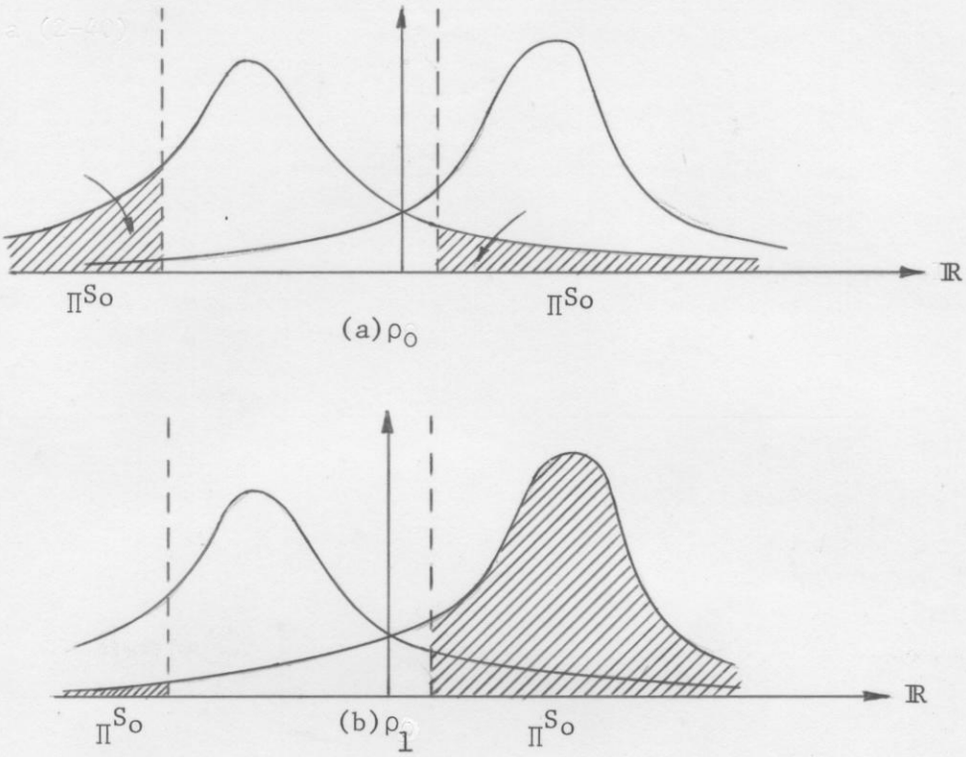


Figura 2.13 - PARTIÇÃO NÃO SIMÉTRICA.

Dada uma partição $\Pi = \{\Pi^s\}$, o procedimento aqui estabelecido indica como realizar uma atribuição para os pesos α_s das classes de confiabilidade, sem indicar contudo como proceder para realizar uma "boa" escolha da partição.

2.5.3.3 - O Algorítmo de Decodificação

O decodificador proposto deve operar sobre a palavra recebida $\underline{r}$, escolhendo a palavra código $\underline{f}$ mais próxima segundo o critério da distância generalizada, i.e., deve escolher uma palavra código $\underline{f} \in C$ tal que $(\forall \underline{f}' \in C)\ D_G(\underline{r},\underline{f}) \leq D_G(\underline{r},\underline{f}')$ (2-53)

Para atingir este fato, é proposta uma regra de decodificação que consiste em estimar a palavra transmitida como sendo a palavra código $\underline{f} \in C$ a qual minimiza a expressão $A(\underline{r},\underline{f}) \triangleq \{\underline{r} \oplus \underline{f}\}.\underline{\alpha}$ ( )

Isto pode ser facilmente demonstrado através do seguinte lema:

Lema 1: Dado a palavra recebida $\underline{r}$, a palavra código $\underline{f}$ mais próxima segundo o critério de distância generalizada é aquela que minimiza $A(\underline{r},\underline{f})$.

Prova - Dada a palavra recebida $\underline{r}$ e uma palavra código $\underline{f}$, a distância genralizada entre elas é dada por

$$D_G(\underline{r},\underline{f}) = \sum_{i=1}^{n} d_G(r_i,f_i) \tag{2-54}$$

Se é observado que $d_G(r_i,f_i) = d_H(r_i,f_i)\ \alpha^{(i)} + \beta_c^{(i)}$, então, é possível reescrever a equação (2-54) como

$$D_G(\underline{r},\underline{f}) = \sum_{i=1}^{n} d_H(r_i,f_i)\alpha^{(i)} + \beta_c^{(i)} \tag{2-55}$$

Por outro lado, o segundo termo do somatório independe da escolha da palavra código, de modo que

$$\underset{\underline{f}}{\text{Min}}\ D_G(\underline{r},\underline{f}) \Rightarrow \underset{\underline{f}}{\text{Min}} \sum_{i=1}^{n} d_H(r_i,f_i)\ \alpha^{(i)} \tag{2-56}$$

Finalmente, observando-se que

$$[\underline{r} \oplus \underline{f}].\underline{\alpha}^{(\ )} = \sum_{i=1}^{n} d_H(r_i,f_i)\alpha^{(i)},$$ a prova é completada.

C.Q.D

É interessante notar que se existe somente uma classe de confiabilidade, então

$$\underset{\underline{f}}{\text{Min}}\ D_G(\underline{r},\underline{f}) \Rightarrow \underset{\underline{f}}{\text{Min}}\ D_H(\underline{r},\underline{f}).$$

Pode ser observado que o decodificador primeiramente efetua uma decisão abrupta determinando uma palavra recebida $\underline{r}$, a qual pode ou não ser uma palavra código. Em seguida, ele determina

um vetor de confiabilidade $\underline{\alpha}^{(\ )}$, associado a esta palavra recebida.

É facilmente verificado pelo lema 2 abaixo que se a palavra recebida $\underline{r}$ resultante do uso de decisão abrupta for uma palavra código, ela própria é a palavra código mais próxima da palavra recebida no sentido de distância generalizada. Segue-se, portanto, que neste caso, o algorítmo resulta em assumir que esta foi a palavra transmitida, o que equivale a desprezar as informações de confiabilidade. No caso contrário, estas informações são realmente utilizadas no processo de decodificação, possibilitando a correção de erros.

Lema 2: Se a palavra recebida $\underline{r}$ é uma palavra código, então a palavra código mais próxima de $\underline{r}$ no sentido de distância generalizada é a própria palavra recebida.

Prova - Foi provado anteriormente (Lema 1) que minimizar $D_G(\underline{r},\underline{f})$ é o mesmo que minimizar $A(\underline{r},\underline{f})$, através da escolha de uma palavra código, ou seja,

$$\min_{\underline{f}} D_G(\underline{r},\underline{f}) \iff \min_{\underline{f}} A(\underline{r},\underline{f}) \qquad (2\text{-}57)$$

É claro que

$$(\forall \underline{f} \varepsilon c)\ A(\underline{r},\underline{f}) = \sum_{i=1}^{n} d_H(r_i,f_i)\alpha^{(i)} \geq 0 \qquad (2\text{-}58)$$

pois, $0 \leq \alpha^{(i)} \leq 1 \quad i = 1,2,3 \ldots,n.$

A conclusão da demonstração decorre do fato que se $\underline{r}$ é palavra código, então

$$\min_{\underline{f}} A(\underline{r},\underline{f}) = A(\underline{r},\underline{r}) = 0 \qquad (2.59)$$

C.Q.D

O lema 2 permite uma simplificação no procedimento de decodificação. Inicialmente uma decisão abrupta é realizada, resul-

tando em uma palavra recebida $\underline{r}$, e a síndrome correspondente a esta palavra recebida é calculada.

1 - Se esta síndrome é nula, o decodificador libera a palavra, admitindo que não ocorreu nenhum erro.

2 - No caso contrário, um vetor de confiabilidade é calculado e o algorítmo atua da maneira já discutida.

2.5.3.4 - Exemplo:

Será apresentada uma aplicação do algorítmo de decodificação descrito, utilizando a treliça associada ao código de bloco binário considerado. Admitindo que a escolha dos pesos das classes de confiabilidade utilizadas esteja de acordo com o que foi proposto (2.5.3.2), e que a partição simétrica foi escolhida de modo a resultar em uma máxima variabilidade dos pesos das classes; a regra de decodificação pode ser anunciada como: "Estime a palavra transmitida como a palavra código para a qual $\sum_{i=1}^{n} d_H(r_i, f_i)\rho^{(i)}$ é mínimo, onde, $\rho^{(i)} = 1 - \phi^{(i)}/1 + \phi^{(i)}$ e $\phi^{(i)}$ é a razão de verossimilhança".

Este procedimento pode ser facilmente implementado com o auxílio de uma treliça. No exemplo, novamente é assumido que o código (5,3,2) é empregado, e que o canal é perturbado por um ruído com distribuição de probabilidade $\mathcal{N}(0,1)$. É admitido ainda que a fonte bipolar transmite +1V ou -1V correspondendo aos símbolos 0 e 1.

Suponha que as amostras recebidas foram (-0.7,-0.2,-0.3,+0.7,+1.1), de modo que $\underline{\rho}^{(\ )} = (0.6, 0.2, 0.3, 0.6, 0.8)$

A palavra recebida é aquela que seria decodificada por decisão abrupta, ou seja, $\underline{r} = (0,0,0,1,1)$.

Usando a treliça, são obtidos os seguintes passos no processo de decodificação:

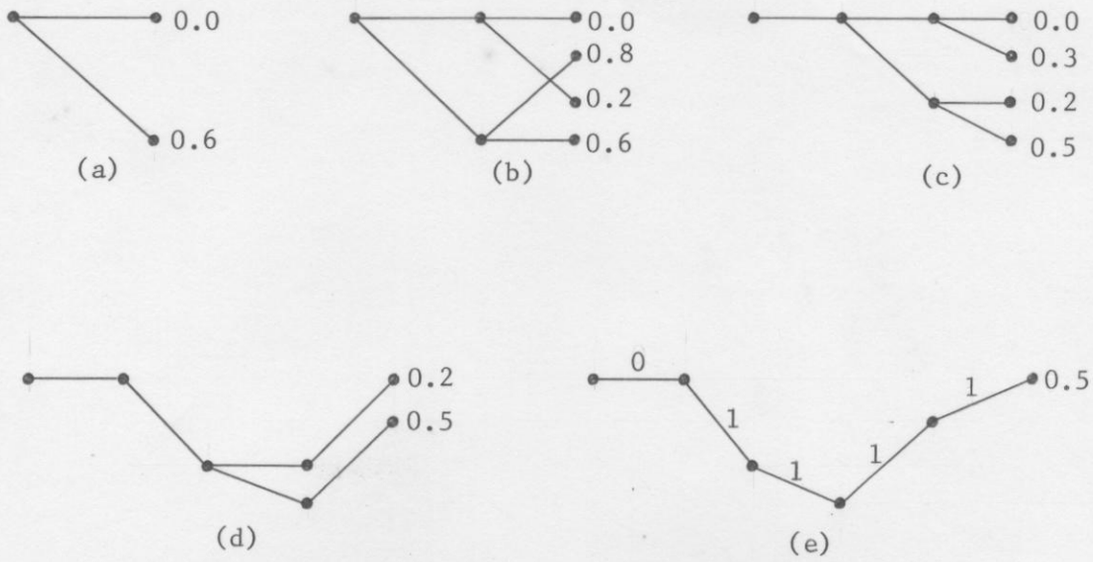


Figura 2.14 - PASSOS NA DECODIFICAÇÃO POR DISTÂNCIA MÍNIMA GENERALIZADA.

A palavra código (1,0,0,1,1) está mais próxima da palavra recebida (0,0,0,1,1) no sentido de distância de Hamming; entretanto, a palavra decodificada foi (0,1,1,1,1), mais próxima no sentido de distância generalizada. Os dígitos corrigidos foram exatamente as posições de mais baixa confiabilidade. Também deve ser observado que a escolha da palavra código (1,0,0,1,1) implicaria em inverter um dígito com confiabilidade relativamente alta. Finalmente, este procedimento apresenta uma vantagem computacional, pois nem sempre é necessário efetuar cálculos, visto que quando $d_H(r_i, f_i) = 0$, nada se adiciona ao somatório.

2.6 - DECODIFICAÇÃO POR DECISÃO SUAVE EMPREGANDO SUBTRELIÇAS

Matis e Modestino |25| introduziram técnicas que permitem uma redução na complexidade da decodificação de códigos de bloco lineares para algorítmos que empregam a treliça gerada pela matriz [H]. Uma das formas proposta para realizar isto é através do uso de

subtreliças bem mais simples que a treliça associada ao código. Se a palavra recebida $\underline{r}$ tem vários coeficientes de confiabilidade com valores grandes, as saídas de decisão abrupta correspondentes podem ser consideradas corretas. A subtreliça é gerada aceitando como corretas estas saídas de decisão abrupta para os símbolos nas k-p posições mais confiáveis, e permitindo todas as possíveis combinações nas posições restantes. É interessante observar que a escolha p=k corresponde ao emprego total da treliça, enquanto que o extremo oposto p=0 corresponde a realizar uma decisão abrupta sem uso da treliça. Valores intermediários de p fornecem as subtreliças desejadas. A aplicação deste procedimento é melhor ilustrada considerando o exemplo apresentado na sec. 2.5.3.4, onde

$$\underline{r} = (0,0,0,1,1) \text{ e } \underline{\alpha}^{(\ )} = (.6,.2,.3,.6,.8).$$

As subtreliças geradas para este caso são mostradas na figura 2.15 (a) e (c).

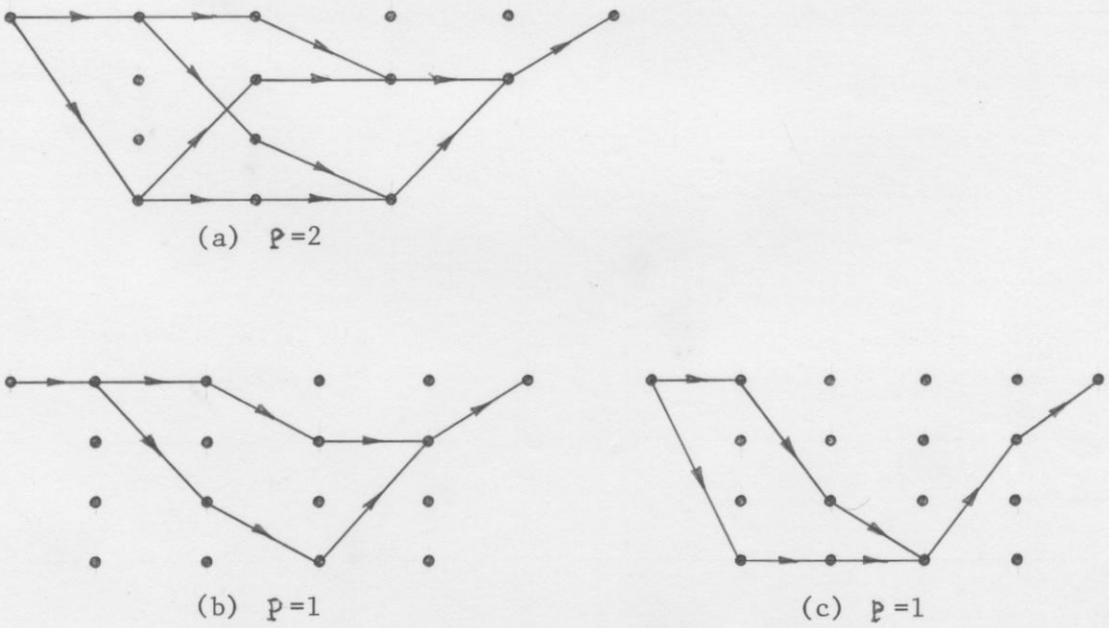


Figura 2.15 - EXEMPLO DE SUBTRELIÇAS PARA UM CÓDIGO DE BLOCO.

O desempenho da decodificação por máxima verossimilhança (descrito na sec. 2.3) empregando subtreliças é analisado para alguns códigos, na referência |25|, através de simulação em computador.

CAPÍTULO III

DECODIFICAÇÃO ÓTIMA DE CÓDIGOS LINEARES

Serão apresentadas duas técnicas ótimas de decodificação de códigos lineares em canais discretos no tempo e sem memória, quando as palavras código são equiprováveis. Ambas fazem uso de decisão suave e são ótimas no sentido que minimizam a probabilidade de erro por símbolo e por bloco, respectivamente. A primeira delas, o algorítmo de Hartmann e Rudolph |18|, faz uso das propriedades de linearidade do código, sendo portanto, aplicável somente para códigos lineares. Ela é particularmente atrativa para códigos corretores de erros com alta eficiência, conforme será visto. A segunda técnica descrita é um algorítmo para maximização da probabilidade a posteriori das palavras código |29|, aplicável a códigos não lineares tão bem quanto para códigos lineares. Quando estes são considerados, o uso da treliça associada na decodificação torna prático seu emprego por simplificar o processo de decodificação, preservando ao mesmo tempo sua otimalidade. O desempenho deste algorítmo é inferior ao desempenho da regra de decodificação de Hartmann-Rudolph com relação a taxa de erros por símbolo, e vice-versa com relação a taxa de erros por palavra. Para ambos os algorítmos será analisado o desempenho assintótico da probabilidade de erro, por bit e por palavra, respectivamente, quando são utilizados códigos de bloco binários e o canal é perturbado por ruído branco gaussiano.

3.1 - O ALGORÍTMO DE HARTMANN E RUDOLPH

3.1.1 - Introdução

A regra de decodificação apresentada pelo algorítmo de Hartmann-Rudolph é ótima, no sentido de que minimiza a probabilidade de erro por símbolo sobre um canal discreto no tempo e sem memória, para qualquer código linear corretor de erros, quando as palavras código são equiprováveis.

A maioria das técnicas de decodificação de códigos corretores de erros são exaustivas no sentido de que toda palavra código é utilizada no processo de decodificação. Estas técnicas não fazem qualquer uso essencial da linearidade do código.* O algorítmo aqui apresentado é também exaustivo, mas no sentido de que toda palavra do código dual é utilizada no processo de decodificação; por isto, na prática, esta regra é usada somente para códigos cujo código dual tem um pequeno número de palavras, i.e., ela é particularmente atrativa para códigos de alta eficiência. Por exemplo, para o código de Hamming binário $C(15,11,3)$ existem 2^{11} palavras código, enquanto que o código dual $C'(15,4,3)$ possui apenas 2^4 palavras código. É concluido, portanto, que a complexidade da regra de decodificação deverá variar inversamente com a eficiência do código. Este algorítmo é de natureza essencialmente estatística e, é aplicável tanto a códigos de bloco quanto a códigos convolucionais. No apêndice A é apresentado um programa em Fortran IV utilizado na investigação do desempenho deste algorítmo para códigos de bloco binários em canais com ruído branco aditivo gaussiano. Os resultados destas simulações serão discutidos no capítulo a seguir.

* Uma maneira de utilizar a linearidade é considerar o uso da treliça associada ao código, descrita na sec.2.3.

3.1.2 - A Regra de Decodificação

A regra de decodificação é apresentada para códigos de bloco lineares, e posteriormente uma extensão para códigos convolucionais será feita.

Seja $\underline{c} = (c_0, c_1, \ldots, c_{n-1})$ uma palavra de um código de bloco linear $C(n,k,d)$ sobre $GF(p)$, e $\underline{c}'_j = (c'_{j0}, c'_{j1}, \ldots, c'_{jn-1})$ a j-ésima palavra do código dual $C'(n,n-k,d')$. A palavra $\underline{c}$ é transmitida através de um canal discreto no tempo e sem memória cujos símbolos r_i são números reais, de modo que a palavra recebida é da forma $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$, com $r_i \in R$. A questão que o algorítmo se propõe a resolver é: dado $\underline{r}$, calcular uma estimativa $\hat{c}_m$ do m-ésimo dígito c_m da palavra transmitida $\underline{c}$, de maneira que a probabilidade de se ter $\hat{c}_m$ igual a c_m seja maximizada. De outra maneira, a estimativa $\hat{c}_m$ é tal que

$$Pr(c_m = \hat{c}_m | \underline{r}) \geq Pr(c_m = s | \underline{r}), \ \forall s \in GF(p). \tag{3-1}$$

A regra de decodificação pode ser assim descrita: Estime o dígito transmitido c_m como sendo o valor de $s \in GF(p)$ que maximiza a expressão

$$A_m(s) = \sum_{t=0}^{p-1} \Theta^{-st} \sum_{j=1}^{p^{n-k}} \left[\prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} \Theta^{-i(c'_{j\ell} - t\delta_{m\ell})} Pr(r_\ell | i) \right] ; \tag{3-2}$$

ou seja, a estimativa $\hat{c}_m$ deve ser tal que $A_m(\hat{c}_m) = \max_{s \in GF(p)} A_m(s)$.

Isto pode ser provado mostrando que $P_r(c_m = s | \underline{r}) = \lambda A_m(s)$, onde λ é uma constante positiva.

Inicialmente, é possível escrever que

$$Pr(c_m = s | \underline{r}) = \sum_{\substack{\underline{c} \in C, \\ c_m = s}} P(\underline{c} | \underline{r}) \tag{3-3}$$

ou

$$Pr(c_m=s|\underline{r}) = \sum_{\substack{\underline{c}\varepsilon C,\\ c_m=s}} P(\underline{r}|\underline{c}) \left[P(\underline{c})/P(\underline{r})\right] \tag{3-4}$$

Considerando que os símbolos de C são elementos de GF(p) e que as palavras código são equiprováveis, tem-se que $P(\underline{c}) = 1/p^k$, e (3-4) pode ser reescrita como

$$P(c_m = s|\underline{r}) = \frac{p^{-k}}{P(\underline{r})} \sum_{\underline{c}\varepsilon C} P(\underline{r}|\underline{c})\delta_{0,(\underline{c}.\underline{e}_m-s)} \tag{3-5}$$

Onde $\delta_{\ell j}$ é o delta de Kronecker,

$$\delta_{\ell j} = \begin{cases} 1, \text{ se } \ell=j \\ 0, \text{ em caso contrário} \end{cases} \quad \text{e} \quad \underline{e}_m \triangleq (\delta_{m0}, \delta_{m1}, \ldots, \delta_{m\,n-1}).$$

Em termos de transformada finita de Fourier, pode ser mostrado que |18|

$$\delta_{0\,x} = p^{-1} \sum_{t=0}^{p-1} \Theta^{tx} \tag{3-6}$$

e

$$P(\underline{r}|\underline{c}) = p^{-n} \sum_{\underline{u}\varepsilon V_n} F(\underline{r},\underline{u})\, \Theta^{\underline{u}.\underline{c}} \tag{3-7}$$

onde

$$F(\underline{r},\underline{u}) = \sum_{\underline{v}\varepsilon V_n} P(\underline{r}|\underline{v})\, \Theta^{-\underline{u}.\underline{v}} \tag{3-8}$$

e $\Theta = \exp(\sqrt{-1}\, 2\Pi/p)$ representa a p-ésima raiz complexa da unidade, e V_n o espaço n-dimensional sobre GF(p).

Substituindo (3-6) e (3-7) em (3-5) e levando em conta que devido as propriedades de ortogonalidade de caracteres de grupo

$$\sum_{\underline{c}\varepsilon C} \Theta^{\underline{v}.\underline{c}} = \begin{cases} p^k, \text{ se } \underline{v} \,\varepsilon\, C' \\ 0\ , \text{ em caso contrário} \end{cases} \tag{3-9}$$

É obtida a expressão

$$P(c_m=s|\underline{r}) = \left[\frac{p^{-n-1}}{P(\underline{r})}\right] \sum_{t=0}^{p-1} \Theta^{-st} \sum_{j=1}^{p^{n-k}} F(\underline{r},\underline{c}'_j - t\underline{e}_m) \qquad (3\text{-}10)$$

Para canais sem memória, (3-8) pode ser reescrita como

$$F(\underline{r},\underline{u}) = \sum_{\underline{v}\varepsilon V_n} \prod_{\ell=0}^{n-1} P(r_\ell|v_\ell)\Theta^{-u_\ell v_\ell} = \prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} P(r_\ell|i)\Theta^{-iu_\ell}, \quad \text{de}$$

modo que se obtém a expressão

$$P(c_m = s|\underline{r}) = \frac{p^{-n-1}}{P(\underline{r})} \cdot \sum_{t=0}^{p-1} \Theta^{-st} \cdot$$

$$\sum_{j=1}^{p^{n-k}} \left[\prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} \Theta^{-i(c'_{j\ell} - t\delta_{m\ell})} P(r_\ell|i)\right] \qquad (3\text{-}11)$$

Definindo $\lambda = p^{-n-1}/P(\underline{r})$, e observando a definição de $A_m(s)$, segue-se que $P(c_m = s|\underline{r}) = \lambda A_m(s)$

C.Q.D.

É interessante fazer uma aplicação deste algorítmo no caso de códigos lineares binários, onde a regra de decodificação se apresenta de uma forma comparativamente mais simples. Neste caso, $p = 2$, e o procedimento para decodificação consiste em escolher a estimativa

$$\hat{c}_m = \begin{cases} 0, & \text{se } A_m(0) > A_m(1) \\ 1, & \text{em caso contrário} \end{cases} \qquad (3\text{-}12)$$

Esta forma comparativa da regra de decisão pode ser apresentada de maneira mais conveniente, quando estabelecida em termos da razão de verossimilhança $\phi_m = Pr(r_m|1)/Pr(r_m|0)$.

Para o caso em questão, a desigualdade $A_m(0) > A_m(1)$ resulta

$$\sum_{t=0}^{1} \sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \sum_{i=0}^{1} (-1)^{-i(c'_{j\ell} - t\delta_{m\ell})} Pr(r_\ell|i) >$$

$$\sum_{t=0}^{1} (-1)^t \sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \sum_{i=0}^{1} (-1)^{-i(c'_{j\ell} - t\delta_m)} Pr(r_\ell|i) \qquad (3\text{-}13)$$

ou seja,

$$\sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \left[Pr(r_\ell|0) + (-1)^{-c'_{j\ell} - \delta_{m\ell}} Pr(r_\ell|1) \right] > 0 \qquad (3\text{-}14)$$

Dividindo ambos os membros de (3-14) pela quantidade positiva $\prod_{\ell=0}^{n-1} Pr(r_\ell|0)$ e usando a definição de razão de verossimilhança, a desigualdade pode ser escrita como

$$\sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \left[1 + \phi_\ell (-1)^{c'_{\ell j} - \delta_{m\ell}} \right] > 0 \qquad (3\text{-}15)$$

Agora dividindo (3-15) pela quantidade positiva $\prod_{\ell=0}^{n-1} (1 + \phi_\ell)$, e utilizando a identidade

$$\frac{1 + \phi_\ell (-1)^{-c'_{j\ell} - \delta_{m\ell}}}{1 + \phi_\ell} = \left(\frac{1 - \phi_\ell}{1 + \phi_\ell} \right)^{c'_{j\ell} \oplus \delta_{m\ell}}$$

, segue-se que a regra de decodificação no caso binário pode ser estabelecida conforme enunciada abaixo:

Escolha $\hat{c}_m = 0$ se $\sum_{j=1}^{2^{n-k}} \prod_{\ell=0}^{n-1} \left(\frac{1 - \phi_\ell}{1 + \phi_\ell} \right)^{c'_{j\ell} \oplus \delta_{m\ell}} > 0$, caso contrário faça $\hat{c}_m = 1$.

Algumas observações devem ser feitas. Em primeiro lugar deve ser notado que quando esta regra de decodificação símbolo-por-símbolo é usada, a palavra estimada $\hat{c}$ na saída do decodificador

não é necessariamente uma palavra código. Em segundo lugar, se códigos de bloco cíclicos são considerados, a regra de decodificação deve ser determinada para o símbolo recebido r_0, e então, os símbolos restantes $r_1,\ldots,r_{n-1}$ podem ser decodificados simplesmente permutando ciclicamente a palavra $\underline{r}$ no registro a deslocamento (acumulador), como ilustrado no exemplo a seguir.

3.1.3 - Exemplos

Será utilizado o código de Hamming (7,4,3) para exemplificar a regra de decodificação binária aplicada a códigos de bloco, que neste caso torna-se:

$$\text{Escolha } \hat{c}_0 = 0 \text{ se } \sum_{j=1}^{8} \prod_{\ell=0}^{6} \left(\frac{1-\phi_\ell}{1+\phi_\ell}\right)^{c'_{j\ell} \oplus \delta_{m\ell}} > 0, \quad \text{e}$$

e $\hat{c}_0 = 1$ em caso contrário.

A matriz de verificação de paridade deste código é

$$[H] = \left|\begin{array}{cccc:ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right|$$

e consequentemente o espaço linha C' será dado por

$$C' = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{matrix}$$

Figura 3.1 - ESPAÇO LINHA DO CÓDIGO DUAL DO CÓDIGO DE HAMMING (7,4,3).

Tomando $\rho_\ell \triangleq (1 - \emptyset_\ell)/(1 + \emptyset_\ell)$, a regra de decodificação consiste em escolher $\hat{c}_0 = 0$ somente se

$$\rho_0 + \rho_1\rho_2\rho_4 + \rho_2\rho_5\rho_6 + \rho_1\rho_3\rho_6 + \rho_3\rho_4\rho_5 + \rho_0\rho_1\rho_2\rho_3\rho_5 + \rho_0\rho_2\rho_3\rho_4\rho_6 + \rho_0\rho_1\rho_4\rho_5\rho_6 > 0.$$

O decodificador correspondente está esquematizado na figura 3.2 exibida abaixo

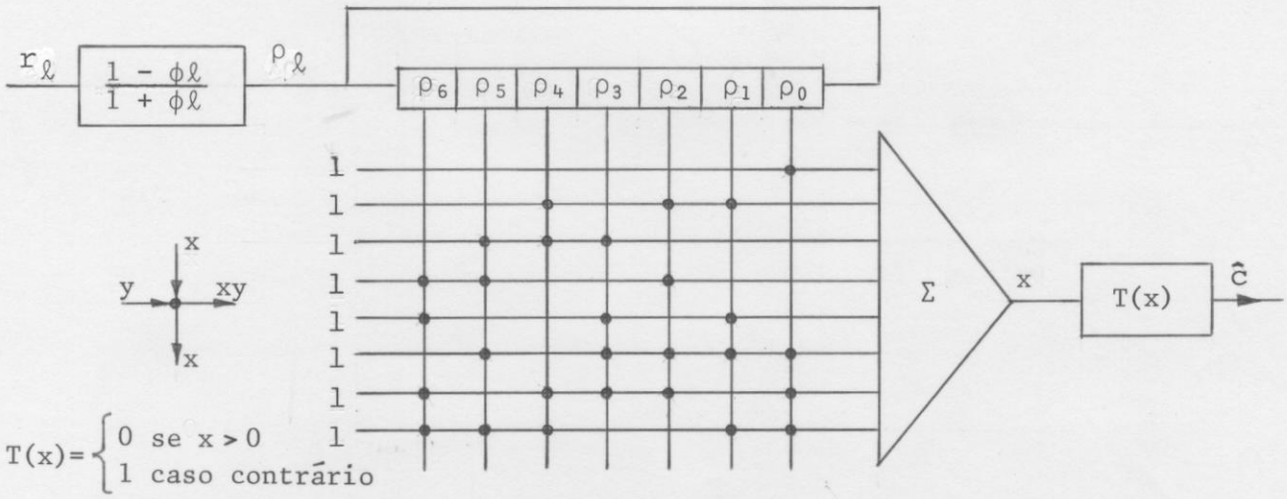


Figura 3.2 - DECODIFICADOR ÓTIMO PARA O CÓDIGO DE BLOCO(7,4,3).

Para ilustrar a aplicação desta regra de decodificação para códigos convolucionais será usado um código convolucional (4,3,3). A regra de decodificação para o símbolo recebido r_0 usando um código convolucional (n_0,k_0,N) será, no caso binário, enunciada como:

$$\text{Escolha } \hat{c}_0 = 0 \text{ se e somente se } \sum_{j=1}^{+\infty} \prod_{\ell=0}^{+\infty} \left(\frac{1-\phi_\ell}{1+\phi_\ell}\right)^{c'_{j\ell} \oplus \delta_{m\ell}} > 0,$$

de outra forma assuma $\hat{c}_0 = 1$.

Naturalmente, existe somente um número finito de termos não nulos na equação acima, dependendo do comprimento da sequência transmitida.

Agora, para exemplificar, as porções iniciais da matriz de verificação de paridade descrita no exemplo 1.5 e do espaço linha C' são mostradas abaixo.

$$H_\infty = \left|\begin{array}{cccccccccccccccccccc}
1 & 1 & 1 & 1 & & & & & & & & & & & & & & & & \\
1 & 0 & 1 & 0 & & 1 & 1 & 1 & 1 & & & & & & & & & & & \\
1 & 1 & 0 & 0 & & 1 & 0 & 1 & 0 & & 1 & 1 & 1 & 1 & & & & & & \\
 & & & & & 1 & 1 & 0 & 0 & & 1 & 0 & 1 & 0 & & 1 & 1 & 1 & 1 \\
 & & & & & & & & & & & \downarrow & & & & & & & \\
 & & & & & & & & & & & \infty & & & & & & &
\end{array}\right|$$

$$C' \quad \begin{array}{cccccccccccccccc}
0 & 0 & \cdots & & & & & & & & & & & & & \\
1 & 1 & 1 & 1 & 0 & \cdots & & & & & & & & & & \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & \cdots & & & & & & \\
0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & \cdots & & & & & & \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & \cdots & & \\
 & & & \downarrow & & & & & & & & \downarrow & & & & \\
 & & & \infty & & & & & & & & \infty & & & &
\end{array}$$

Figura 3.3 - ESPAÇO LINHA DO CÓDIGO DUAL DO CÓDIGO CONVOLUCIONAL (4,3,3).

Como antes, $\rho_\ell \triangleq (1 - \alpha_\ell)/(1 + \alpha_\ell)$, de modo que a regra de decodificação será expressa por:

$$\hat{c}_0 = 0 \text{ se e somente se } \rho_0 + \rho_1\rho_2\rho_3 + \rho_2\rho_4\rho_5\rho_6\rho_7 + \rho_0\rho_1\rho_3\rho_4\rho_5\rho_6\rho_7 + \\ + \ldots > 0, \text{ em caso contrário faça } \hat{c}_0 = 1.$$

O diagrama do decodificador é então mostrado na figura 3.4, tomando forma de diagrama de treliça para o código dual $c'(4,1,3)$ com as posições c'_{j0} nos rótulos das malhas complementadas*. (Em geral, para decodificar r_m, as posições c'_{jm} devem ser complementadas).

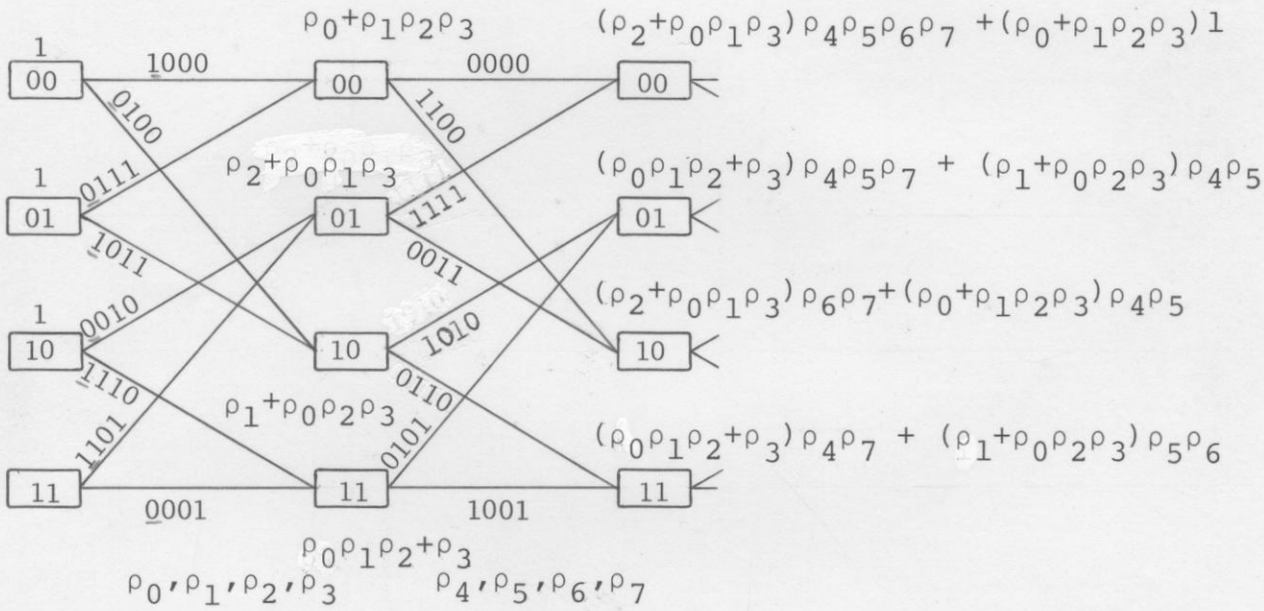


Figura 3.4 - DECODIFICADOR ÓTIMO PARA O CÓDIGO CONVOLUCIONAL (4,3,3).

Desde que diferentes unidades de memória devem ser empregadas para cada símbolo a ser decodificado, a quantidade de memória requerida por este tipo de decodificador cresce linearmente com o comprimento da sequência transmitida. O mesmo também é verdade para o algorítmo de Viterbi aplicado para decodificação de códigos convolucionais.

* Veja figura 1.4, diagrama de treliça para o código convolucional (4,1,3).

3.1.4 - Desempenho Assintótico do Algorítmo

Expressões assintóticas para a probabilidade de erro por bit na decodificação de códigos de bloco binários lineares em canal com ruído branco aditivo gaussiano foram demonstradas |19|.

A regra de decodificação ótima bit-a-bit para o m-ésimo dígito de um código $C(n,k,d)$ linear e binário consiste em escolher $\hat{c}_m = s$, onde $s \varepsilon GF(2)$ e maximiza $p(\hat{c}_m = s|\underline{r})$. Desta forma a probabilidade de erro por bit é dada por

$$P_{bit} = P\left[p(\hat{c}_m = c_m|\underline{r}) \leq p(\hat{c}_m \neq c_m|\underline{r})\right] \tag{3-16}$$

Não há perda de generalidade em assumir que a palavra transmitida foi a énupla toda nula, desde que o código é linear e o ruído apresenta simetria. Para facilitar a dedução, é considerado que a palavra toda nula foi transmitida. Quando $\underline{c} = \underline{0}$ é transmitida, a m-ésima componente da palavra recebida $\underline{r}$ é $r_m = \sqrt{E} + e_m$, onde E é a energia do sinal por bit e e_m uma amostra de ruído de um processo gaussiano com densidade espectral de potência unilateral η_0 watts/hertz. A SNR para este canal é $\gamma = E/\eta_0$, ou em termos dos bits de informação transmitidos, $\gamma_b = E_b/\eta_0 = \gamma_n/k = \gamma/R$. A m-ésima componente da palavra recebida $\underline{r}$ será decodificada incorretamente se e somente se

$$P(c_m = 0|\underline{r}) \leq P(c_m = 1|\underline{r}) \tag{3-17}$$

onde $\underline{r} = (\sqrt{E} + e_0, \ldots, \sqrt{E} + e_{n-1})$.

Em outras palavras,

$$P_{bit} = P\left[\sum_{\underline{c}\varepsilon s_0} P(\underline{c}|\underline{r}) \leq \sum_{\underline{c}\varepsilon s_1} P(\underline{c}|\underline{r})\right] \tag{3-18}$$

onde $s_i \triangleq \{\underline{c} \in C | c_m = i\}, \quad i = 0,1.$

Desde que o canal é sem memória e as palavras código são equiprováveis, (3-18) pode ser escrita como

$$P_{bit} = P\left[\sum_{\underline{c}\in s_0} \prod_{\ell=0}^{n-1} P(r_\ell|c_\ell) \leq \sum_{\underline{c}\in s_1} \prod_{\ell=0}^{n-1} P(r_\ell|c_\ell)\right] \tag{3-19}$$

Como $r_\ell = e_\ell + (-1)^{c_\ell}\sqrt{E}$ é uma variável aleatória real e contínua com densidade de probabilidade

$$g(r_\ell|c_\ell) = \frac{1}{\sqrt{\pi\eta_0}} \exp\left[-\frac{(r_\ell - (-1)^{c_\ell}\sqrt{E})}{\eta_0}\right]$$ e o produto

$\prod_{\ell=0}^{n-1} g(r_\ell|0)$ é positivo, segue-se que

$$P_{bit} = P\left[\sum_{\underline{c}\in s_0} \prod_{\ell=0}^{n-1} \frac{g(r_\ell|c_\ell)}{g(r_\ell|0)} \leq \sum_{\underline{c}\in s_1} \prod_{\ell=0}^{n-1} \frac{g(r_\ell|c_\ell)}{g(r_\ell|0)}\right] \tag{3-20}$$

ou seja,

$$P_{bit} = P\left[\sum_{\underline{c}\in s_0} \exp(-\underline{r}.\underline{c}\sqrt{R\gamma_b/\eta_0}) \leq \sum_{c\in s_1} \exp(-\underline{r}.\underline{c}\sqrt{R\gamma_b/\eta_0})\right] \tag{3-21}$$

Para análise do comportamento assintótico de P_{bit}, denotado $AB_{\gamma b\to 0}(P_{bit})$ e $AB_{\gamma b\to\infty}(P_{bit})$, η_0 será fixado e E_b variado de modo que γ_b varie.

3.1.4.1 - Caso de Baixa Relação Sinal-Ruído

Para $x << 1$, $\exp x \simeq 1 + x$, de modo que numa pequena vizinhança de zero é possível escrever

$$AB_{\gamma b\to 0}(P_{bit}) \approx P\left[\sum_{\underline{c}\varepsilon s_0} (1 - \underline{r}.\underline{c}\ 4\sqrt{R\gamma_b/\eta_0}) \leq \sum_{\underline{c}\varepsilon s_1} (1 - \underline{r}.\underline{c}\ 4\sqrt{R\gamma_b/\eta_0})\right] \tag{3-22}$$

Então, como $|s_0| = |s_1|$,

$$AB_{\gamma b\to 0}(P_{bit}) \approx P\left[\sum_{\underline{c}\varepsilon s_0} - \underline{r}.\underline{c}\ 4\sqrt{R\gamma_b/\eta_0} \leq \sum_{\underline{c}\varepsilon s_1} \underline{r}.\underline{c}\ 4\sqrt{R\gamma_b/\eta_0}\right] \tag{3-23}$$

Desde que s_0 é um código linear binário $(n,k-1)$, segue que se os vetores de s_0 são arranjados como linhas de uma matriz M_0, então cada coluna será toda nula ou conterá 2^{k-2} zeros e 2^{k-2} 1's. Arranjando o conjunto de vetores de s_1 como linhas' de uma matriz M_1, as colunas de M_1 que correspondam a colunas toda nula em M_0 serão toda um, enquanto que as demais colunas terão 2^{k-2} zeros e 2^{k-2} 1's.

Usando este fato, é possível escrever

$$AB_{\gamma b\to 0}(P_{bit}) \approx P\left[\sum_{\ell=j_1}^{j_\theta} r_\ell \leq 0\right] \tag{3-24}$$

onde $j_1, j_2, \ldots, j_\theta$ são colunas nulas de M_0.

Desde que $r_\ell \sim \mathcal{N}(\sqrt{E}, \eta_0/2)$ para $\ell = j_1, \ldots, j_\theta$,

$$P\left[\sum_{\ell=j_1}^{j_\theta} r_\ell \leq 0\right] = \frac{1}{\sqrt{2\Pi}} \int_{\sqrt{2\theta R\gamma_b}}^{+\infty} \exp(-x^2/2)\,dx \tag{3-25}$$

De (3-24) e (3-25) segue-se a expressão assintótica desejada, $AB_{\gamma b\to 0}(P_{bit}) \approx Q(\sqrt{2\theta R\gamma_b})$.

Se o código dual de C tem distância mínima maior que 2, então $\theta = 1$, e o comportamento assintótico pode ser descrito pe

la equação $AB_{\gamma b \to 0}(P_{bit}) \approx Q(\sqrt{2R\gamma_b})$.

3.1.4.2 - Caso de Alta Relação Sinal-Ruído

Desde que $\underline{0} \in s_0$, $\sum_{\underline{c} \in s_0} \exp(-\underline{r}.\underline{c}\ 4\sqrt{R\gamma_b/\eta_0}) \geq 1$.

Se $\underline{r}.\underline{c} > 0$ para todo $\underline{c} \in s_1$, então quando $\gamma_b \to \infty$

$$P\left[1 \leq \sum_{c \in s_0} \exp(-\underline{r}.\underline{c}\ 4\sqrt{R\gamma_b/\eta_0}) \leq \sum_{\underline{c} \in s_1} \exp(-\underline{r}.\underline{c}\ 4\sqrt{R\gamma_b/\eta_0})\right] \to 0 \qquad (3\text{-}26)$$

consequentemente, (3-21) é cotada superiormente pelo "union bound " quando γ_b cresce, i.e.,

$$AB_{\gamma b \to \infty}(P_{bit}) \leq P\left[\bigcup_{\underline{c} \in s_1} \underline{r}.\underline{c} \leq 0\right] \leq \sum_{\underline{c} \in s_1} P(\underline{r}.\underline{c} \leq 0) \qquad (3\text{-}27)$$

Pode ser mostrado |19| que esta cota é apertada para valores de γ_b suficientemente grandes, de modo que

$$AB_{\gamma b \to \infty}(P_{bit}) \approx \sum_{c \in s_1} P(\underline{r}.\underline{c} \leq 0) \qquad (3\text{-}28)$$

Assim,

$$P(\underline{r}.\underline{c} \leq 0) = \frac{1}{\sqrt{2\Pi}} \int_{\sqrt{2RW(\underline{c})\gamma_b}}^{+\infty} \exp(-x^2/2)\,\delta x = Q(\sqrt{2RW(\underline{c})\gamma_b}) \qquad (3\text{-}29)$$

Portanto, substituindo (3-29) em (3-28), tem-se:

$$AB_{\gamma b \to \infty}(P_{bit}) \approx \sum_{\underline{c} \in s_1} Q(\sqrt{2R\ W(\underline{c})\gamma_b}) \qquad (3\text{-}30)$$

Se W_m é o menor peso das palavras código em s_1 e $N(W_m)$ o número de palavras código de peso W_m, então

$AB_{\gamma b\to\infty}(P_{bit}) \approx N(W_m)\ Q(\sqrt{2R\ W_m\ \gamma_b})$, isto porque para valores suficientemente grandes de x, $Q(x) \approx 1/x\sqrt{2\Pi}\ \exp(-x^2/2)$, de maneira que somente as palavras código de peso W_m contribuem de forma significante no somatório. Deve ser também observado que se o código é cíclico e tem distância mínima d, então $W_m = d$ e

$$AB_{\gamma b\to\infty}(P_{bit}) \approx N(d)\ Q(\sqrt{2Rd\ \gamma_b})$$

3.2 - ALGORÍTMO PARA MAXIMIZAÇÃO DA PROBABILIDADE A POSTERIORI

3.2.1 - Introdução

Uma outra regra de decodificação será a seguir apresentada a qual é também ótima, mas no sentido de que para qualquer código corretor de erros minimiza a probabilidade de erro por palavra sobre qualquer canal discreto no tempo e sem memória, quando as palavras código são equiprováveis. Esta regra implica em maximizar a probabilidade a posteriori das palavras código. É sabido que exceto para poucos canais clássicos não é normalmente muito claro como deve ser processada a palavra recebida de modo a maximizar a probabilidade a posteriori. No que segue é estabelecido um procedimento para realizar tal maximização. Deve ser observado que ao contrário do algorítmo de Hartmann-Rudolph, a decodificação resulta sempre em uma palavra código quando este procedimento é empregado. Será também mostrado que a regra de decodificação para o caso binário pode ser enunciada como:

Escolher a palavra código $\underline{c}$ que maximiza a expressão

$$\sum_{\ell=0}^{n-1} c_\ell \ell n\ \emptyset_\ell .$$

Se dado a palavra recebida $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$ for definido um vetor $\underline{r}^*$ calculado a partir de $\underline{r}$;

$\underline{r}^* = (\ell og\ \emptyset_0, \ell og\ \emptyset_1, \ldots, \ell og\ \emptyset_{n-1})$, onde $\emptyset_j = Pr(r_j|1)/Pr(r_j|0)$, o

decodificador deve escolher a palavra código $\underline{c}$ que maximiza $\underline{c}.\underline{r}^*$, i.e., $\hat{\underline{c}}.\underline{r}^* = \underset{\underline{c}}{\text{Max}}\ \underline{c}.\underline{r}^*$. Isto torna possível uma interpretação interessante, observando que este procedimento generaliza a decodificação por correlação para códigos de bloco sobre canais com ruído aditivo gaussiano, uma das técnicas ótimas conhecidas no sentido de que minimiza a probabilidade de erro por bloco quando as palavras código são equiprováveis. Deste modo, é plausível interpretar esta decodificação como sendo uma forma de "correlação generalizada".

Quando códigos lineares são usados, o emprego da decodificação usando treliça faz este receptor interessante por simplificar as operações de decodificação, retendo entretanto, a otimalidade. É importante observar que este algorítmo de decodificação é atrativo tanto para códigos com alta eficiência, quanto para os de baixa eficiência. Se o código utilizado é de baixa eficiência, então a decodificação pode ser feita exaustivamente, i.e., utilizando todas palavras código no processo de decodificação. Ele também é de particular utilidade na decodificação de códigos com altas taxas, visto que a complexidade da treliça é função do número de dígitos de paridade.

Se ao invés de empregado como na forma descrita acima o algorítmo for utilizado quantizando em 2^Q regiões as amostras recebidas, o resultado é uma pequena degradação no desempenho, porém, facilitando sobremaneira a implementação do decodificador, como mostrado a seguir. A decodificação agora é feita através do seguinte procedimento:

1 - vetor recebido $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$ com $r_i \in R$

2 - vetor quantizado $\underline{r}' = (r'_0, r'_1, \ldots, r'_{n-1})$ com $r'_i \in R_Q$

3 - vetor ℓog razão de verossimilhança

$$\underline{r}^* = (r^*_0, r^*_1, \ldots, r^*_{n-1}) \quad \text{com} \quad r^*_i \in R_\emptyset$$

R = {espaço das possíveis saídas do canal ruidoso}

R_Q = {espaço das possíveis regiões de quantização}

$R_\emptyset$ = {espaço dos valores do ℓog razão de verossimilhança}

É então utilizada a treliça associada ao código para maximizar a "correlação" $\underline{c}.\underline{r}^* = \sum_{\ell=0}^{n-1} c_\ell \, r_\ell^*$.

O algorítmo será deduzido para códigos de bloco com alfabeto de símbolos p-ários, supondo palavras código equiprováveis transmitidas em um canal discreto no tempo e sem memória, e posteriormente uma extensão para códigos convolucionais será feita.

3.2.2 - A Regra de Decodificação

Seja $\underline{c} = (c_0, c_1, \ldots, c_{n-1})$ uma palavra código de um código de bloco $C(n,k,d)$ com símbolos em $GF(p)$. Ao receber a palavra $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$, o receptor deve decidir pela palavra código $\underline{c}$ que maximiza a expressão

$$\prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} P(r_\ell | i) \sum_{j=0}^{p-1} \Theta^{j(c_\ell - i)} \tag{3-31}$$

onde $\Theta = \exp|2\Pi\sqrt{-1}/p|$ representa a p-ésima raiz complexa da unidade.

Será provado que a palavra código que maximiza (3-31) também maximiza a probabilidade a posteriori $P(\underline{c}|\underline{r})$. A probabilidade $P(\underline{c}|\underline{r})$ pode ser determinada por

$$P(\underline{c}|\underline{r}) = P(\underline{c}) \; P(\underline{r}|\underline{c})/P(\underline{r}) \tag{3-32}$$

e como por hipótese as palavras são equiprováveis, $P(\underline{c}) = p^{-k}$, de maneira que

$$P(\underline{c}|\underline{r}) = P^{-k} \; P(\underline{r}|\underline{c})/P(\underline{r}) \tag{3-33}$$

Em termos de transformada finita de Fourier, a probabilidade $P(\underline{r}|\underline{c})$ pode ser escrita como

$$P(\underline{r}|\underline{c}) = p^{-n} \sum_{\underline{u}\varepsilon V_n} F(\underline{r},\underline{u})\,\Theta^{\underline{u}.\underline{c}} \tag{3-34}$$

onde

$$F(\underline{r},\underline{u}) = \sum_{\underline{v}\varepsilon V_n} P(\underline{r}|\underline{v})\,\Theta^{-\underline{u}.\underline{v}} \tag{3-35}$$

Como são considerados apenas canais sem memória,

$$F(\underline{r},\underline{u}) = \sum_{\underline{v}\varepsilon V_n} \prod_{\ell=0}^{n-1} P(r_\ell|v_\ell)\,\Theta^{-u_\ell . v_\ell}$$

$$= \prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} P(r_\ell|i)\,\Theta^{-iu_\ell} \tag{3-36}$$

Substituindo as expressões (3-36) e (3-34) em (3-33) resulta

$$P(\underline{c}|\underline{r}) = [p^{-k-n}/P(\underline{r})] \sum_{\underline{u}\varepsilon V_n} \prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} P(r_\ell|i)\,\Theta^{u_\ell(c_\ell-i)}$$

$$= [\lambda_1] \prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} P(r_\ell|i) \sum_{j=0}^{p-1} \Theta^{j(c_\ell-i)} \tag{3-37}$$

De modo que maximizar (3-31) pela escolha de uma palavra código é o mesmo que maximizar a probabilidade a posteriori das palavras código.

C.Q.D.

Será feita a seguir uma aplicação deste algorítmo para o caso binário, i.e, quando $p=2$.

$$P(\underline{c}|\underline{r}) = [\lambda_1] \prod_{\ell=0}^{n-1} \left[P(r_\ell|0)(1+\Theta^{c_\ell}) + P(r_\ell|1)(1-\Theta)^{c_\ell}\right] \tag{3-38}$$

Mais uma vez é conveniente exprimir os resultados em termos da razão verossimilhança $\phi_\ell = Pr(r_\ell|1)/Pr(r_\ell|0)$, e de $\rho_\ell \triangleq (1 - \phi_\ell)/(1 + \phi_\ell)$. Isto resulta em

$$P(\underline{c}|\underline{r}) = \left[2^{-k-n}/P(\underline{r})\right] \prod_{\ell=0}^{n-1} \left[(1 + \Theta^{c_\ell}) + \phi_\ell(1 - \Theta^{c_\ell})\right].P(r_\ell|0)$$

Tomando $\lambda = \left[2^{-k-n}/P(\underline{r})\right] \prod_{\ell=0}^{n-1} P(r_\ell|0)$, tem-se que

$$P(\underline{c}|\underline{r}) = \lambda \prod_{\ell=0}^{n-1} \left[(1 + \phi_\ell) + \Theta^{c_\ell}(1 - \phi_\ell)\right]$$

$$= \lambda \prod_{\ell=0}^{n-1} \left[1 + \rho_\ell \, \Theta^{c_\ell}\right].(1 + \phi_\ell) \qquad (3\text{-}39)$$

Contudo, $\Theta^{c_\ell} = \exp \frac{\sqrt{-1}\, 2\Pi}{2} \cdot c_\ell = \begin{cases} 1, & \text{se } c_\ell = 0 \\ -1, & \text{se } c_\ell = 1 \end{cases}$

ou seja, $\Theta^{c_\ell} = (-1)^{c_\ell}$, de forma que (3-39) pode ser reescrita como

$$P(\underline{c}|\underline{r}) = \lambda \prod_{\ell=0}^{n-1} \left[1 + \rho_\ell(-1)^{c_\ell}\right] (1 + \phi_\ell) \qquad (3\text{-}40)$$

Usando a identidade $1 + \rho_\ell(-1)^{c_\ell} = \dfrac{2\,\phi_\ell^{c_\ell}}{1+\phi_\ell}$, obtém-se

$P(\underline{c}|\underline{r}) = 2\lambda \prod_{\ell=0}^{n-1} \phi_\ell^{c_\ell}$, e λ independe de $\underline{c}$.

Como o logarítmo é uma função monotônica crescente, a maximização de $P(\underline{c}|\underline{r})$ no caso binário pode ser realizada escolhendo a palavra código $\underline{c} = (c_0,\ldots,c_{n-1})$ que maximiza a expressão $\sum_{\ell=0}^{n-1} c_\ell \, \ell n \phi_\ell$.

3.2.3 - Aplicações para Códigos Lineares

É fácil verificar que no caso da transmissão de sinais polares em um canal com ruído aditivo gaussiano, a regra de decodificação consiste em maximizar a quantidade $\underline{c}.\underline{r} = \sum_{\ell=0}^{n-1} c_\ell r_\ell$ pela escolha de uma palavra código $\underline{c} = (c_0, c_1, \ldots, c_{n-1})$. Se os sinais transmitidos são pulsos liga-desliga de amplitude $\sqrt{E}$ volts ou 0 volts, então $\sum_{\ell=0}^{n-1} c_\ell (r_\ell - \sqrt{E}/2)$ deve ser maximizada pela escolha de $\underline{c}$.

Numa aplicação particular será obtida a regra ótima de decodificação quando são empregados detetores de envoltória. Considere que o sinal é transmitido através de pulsos de RF liga-desliga (ASK) em um canal com ruído aditivo branco gaussiano. É assumido que a envoltória da forma de onda recebida é amostrada em intervalos bastante distantes de modo que seja razoável supor que as amostras são estatisticamente independentes. O sinal recebido é

$$Z(t) = A_i \cos\omega_{0t} + x_c(t) \cos\omega_{0t} - x_s(t) \operatorname{sen} \omega_{0t} \qquad (3\text{-}41)$$

onde $A_0 = 0$ e A_1 = constante, e o ruído está escrito na forma de banda estreita |33| de modo que $x_c(t)$ e $x_s(t)$ são variáveis aleatórias iid gaussianas.

A saída $r(t)$ do detetor de envoltória é dada por

$$r(t) = \{[A_i + x_c(t)]^2 + x_s^2(t)\}^{1/2} \quad i \varepsilon \{0,1\} \qquad (3\text{-}42)$$

A densidade de probabilidade da envoltória em um instante de tempo t |33| tem a distribuição de Rayleigh na ausência de sinal, e uma distribuição de Rice quando o sinal está presente, ou seja,

$$P(r_\ell|0) = \begin{cases} \dfrac{r_\ell}{\sigma^2} \exp(-r_\ell^2/2\sigma^2), & r_\ell \geq 0 \\ 0 & , \quad r_\ell < 0 \end{cases} \tag{3-43}$$

e

$$P(r_\ell|1) = \begin{cases} \dfrac{r_\ell}{\sigma^2} \exp - \left(\dfrac{r_\ell^2 + A_1^2}{2\sigma^2}\right) I_0 \left(\dfrac{A_1 \, r_\ell}{\sigma^2}\right), & r_\ell \geq 0 \\ 0 & , \quad r_\ell < 0 \end{cases} \tag{3-44}$$

onde $\sigma^2 = E\{n_c^2(t)\} = E\{n_s^2(t)\}$ é a potência do ruído, e $I_0(x)$ é a função de Bessel modificada de primeira espécie e de ordem zero .

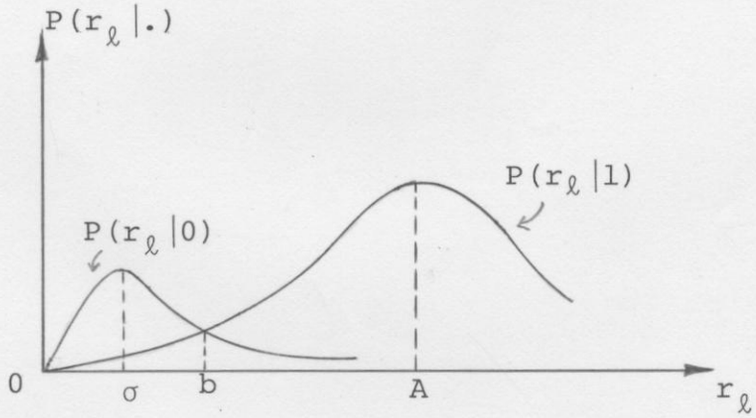


Figura 3.5 - DENSIDADES DE PROBABILIDADE DA ENVOLTÓRIA.

A razão de verossimilhança é dada pela expressão

$$\emptyset_\ell = \frac{P(r_\ell|1)}{P(r_\ell|0)} = \exp(-A_1^2/2\sigma^2) \; I_0(r_\ell A_1/\sigma^2) \tag{3-45}$$

Definindo a relação sinal/ruído como $\gamma = A_1^2/2\sigma^2$, o valor do limiar b pode ser encontrado resolvendo a equação $\exp(-\gamma) I_0(b_0\sqrt{2\gamma}) = 1$, cuja solução pode ser obtida através da excelente aproximação |34|

$$b_0 = \sqrt{2 + \gamma/2} = b/\sigma \tag{3-46}$$

Uma análise do comportamento assintótico da regra de decodificação para baixa e alta relação sinal/ruído será realizada a seguir.

3.2.3.1 - Baixa Relação Sinal/Ruído: $\gamma << 1$

Neste caso é quase sempre verdade que |29|

$$I_0(r_\ell A_1/\sigma^2) \approx \exp(r_\ell^2 A_1^2/4\sigma^2) \tag{3-47}$$

portanto,

$$\ln \phi_\ell \approx \frac{A_1^2}{2\sigma^2} \left[\frac{r_\ell^2}{2\sigma^2} - 1\right] \tag{3-48}$$

e a regra de decodificação consiste em escolher a palavra código $\underline{c} = (c_0,\ldots,c_{n-1})$ que maximiza a expressão

$$\sum_{\ell=0}^{n-1} c_\ell \left[\frac{r_\ell^2}{2\sigma^2} - 1\right] \tag{3-49}$$

Para ilustrar a aplicação desta regra de decodificação, será considerado o código de bloco binário (3,2,2) cuja treliça associada está apresentada abaixo

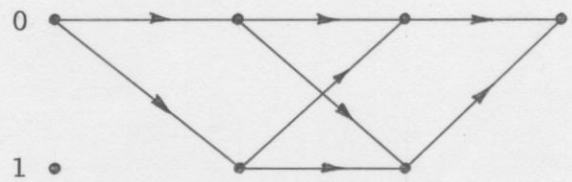


Figura 3.6 - TRELIÇA ASSOCIADA AO CÓDIGO LINEAR (3,2,2).

É assumido que a ênupla recebida é $\underline{r} = (1.7, .1, .5)$, e que o ruído tem média zero e variância unitária. Os passos no processo de decodificação estão mostrados na figura 3.7, resultando na palavra código (0,0,0).

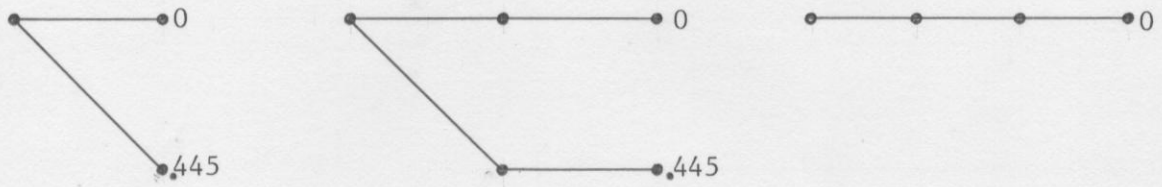


Figura 3.7 - PASSOS NA DECODIFICAÇÃO PARA O CÓDIGO (3,2,2).

Deve ser observado que o emprego de decisão abrupta resultaria na palavra (1,0,0), uma vez que os símbolos r_ℓ são comparados com o limiar $b = \sigma\sqrt{2 + \gamma/2} \approx \sigma\sqrt{2}$. O erro corrigido foi exatamente na amostra mais próxima ao limiar ($r_0 = 1.7$), ou seja, a amostra de menor confiabilidade.

3.2.3.2 - Alta Relação Sinal/Ruído

Neste caso a distribuição de Rice pode ser aproximada por uma distribuição gaussiana |6|

$$P(r_\ell|1) \approx \frac{1}{\sqrt{2\Pi\sigma^2}} \exp - \frac{\left[r_\ell - A_1\right]^2}{2\sigma^2} \qquad (3\text{-}50)$$

De modo que o ℓog da razão de verossimilhança pode ser dado aproximadamente por

$$\ell n\phi_\ell \approx \frac{A_1}{\sigma^2}\left[(r_\ell - \frac{A_1}{2}) - \frac{\sigma^2}{A_1}(\ell n r_\ell - \ell n\sigma_0)\right] \qquad (3\text{-}51)$$

onde $\sigma_0 = \sigma/\sqrt{2\Pi}$.

Portanto, a palavra código escolhida deve maximizar a expressão (3-52) abaixo

$$\sum_{\ell=0}^{n-1} c_\ell \left[(r_\ell - A_1/2) - \frac{\sigma^2}{A_1}(\ell n r_\ell - \ell n\sigma_0)\right] \qquad (3\text{-}52)$$

A extensão para aplicação desta regra na decodificação de códigos convolucionais (n_0, k_0, N) consiste, no caso binário, em escolher o caminho da treliça que maximiza a expressão $\sum_{\ell=0}^{+\infty} c_\ell \ell n\phi_\ell$, onde novamente o número de termos não nulos depende do comprimento da sequência transmitida.

O algorítmo de Viterbi para decodificação de códigos convolucionais pode ser facilmente modificado para fazer uso desta regra de decodificação, e a melhoria provida pelo uso desta técnica de decisão suave encontra-se analisada na referência |1|.

3.2.4 - Análise do Comportamento Assintótico

Será deduzida uma expressão assintótica para a probabilidade de erro por palavra na decodificação ótima de códigos de bloco lineares binários, usando decisão suave, para canal com ruído branco aditivo gaussiano, quando a relação sinal/ruído é elevada. A expressão obtida é função da distribuição de pesos das palavras código e da relação sinal/ruído.

Aqui a atenção será restrita aos códigos de bloco lineares binários com palavras código equiprováveis transmitidas sobre um canal com ruído branco aditivo gaussiano através de pulsos liga-

desliga. É assumido que o código $C(n,k,d)$ considerado tem palavras código denotadas por $\underline{c}_i$, $i = 0,1,\ldots,2^k-1$, onde $\underline{c}_i=(c_{i0},c_{i1},\ldots,c_{in-1})$.

A regra ótima de decodificação é escolher $\hat{\underline{c}} = \underline{c}_i \in C$ que maximiza $P(\hat{\underline{c}} = \underline{c}_i|\underline{r})$, onde $\hat{c}$ é a estimativa da palavra código transmitida e $\underline{r}$ é a palavra recebida. Como visto, isto pode ser realizado maximizando $\sum_{\ell=0}^{n-1} c_{i\ell} \ln \phi_\ell$, onde $\phi_\ell=P(r_\ell|1)/P(r_\ell|0)$ é a razão de verossimilhança. Novamente a dedução dos resultados é simplificada assumindo que a palavra toda nula foi transmitida.

Conhecida a densidade espectral de potência unilateral do ruído $\eta_0/2$, as componentes $\{n_\ell\}$ do vetor de ruído $\underline{n}$ são variáveis aleatórias iid gaussianas, $n_\ell \sim \mathcal{N}(0,\eta_0/2)$. Novamente será utilizada a relação sinal/ruído do canal $\gamma = E/\eta_0$, e a relação sinal/ruído por bit de informação transmitido $\gamma_b = E_b/\eta_0 = \gamma/R$.

Das considerações acima segue-se que a probabilidade de erro por bloco é

$$P_{bloco} = P[\varepsilon \mid \underline{c}=\underline{0}] \tag{3-53}$$

Dado que a ênupla toda zero $\underline{c} = \underline{0}$ foi transmitida, haverá erro na decodificação se e somente se $\sum_{\ell=0}^{n-1} c_{i\ell} \ln \phi_\ell > 0$ para alguma palavra código $\underline{c}_i$, $i = 1,2,\ldots,2^k-1$.

Como os sinais transmitidos são pulsos de amplitude $\sqrt{E}$ volts ou 0 volts, e o canal é gaussiano, segue-se que

$$P(r_\ell|i) = \frac{1}{\sqrt{\pi\eta_0}} \exp\left[-\frac{(r_\ell - i\sqrt{E})^2}{\eta_0}\right] \quad i = 0,1 \tag{3-54}$$

e portanto,

$$\sum_{\ell=0}^{n-1} c_{i\ell} \ln\phi_\ell = \frac{\sqrt{E}}{\eta_0/2} \sum_{\ell=0}^{n-1} c_{i\ell}\left(r_\ell - \frac{\sqrt{E}}{2}\right) \tag{3-55}$$

Como é suposto que a palavra toda zero foi a palavra transmitida, o vetor recebido $\underline{r} = \underline{c} + \underline{n} = (r_0, r_1, \ldots, r_{n-1})$ tem sua ℓ-ésima componente dada por $r_\ell = n_\ell$.

Se são definidos eventos A_i; $i = 1,2,\ldots,2^k-1$,

$$A_i \triangleq \{ \sum_{\ell=0}^{n-1} c_{i\ell}\, n_\ell > \frac{\sqrt{E}}{2} \sum_{\ell=0}^{n-1} c_{i\ell} \} \tag{3-56}$$

então é possível reescrever (3-53) como

$$P_{bloco} = P(\bigcup_{i=1}^{2^k-1} A_i) \tag{3-57}$$

Utilizando o "union bound", obtém-se a expressão:

$$P_{bloco} \leq \sum_{i=1}^{2^k-1} P(A_i) \tag{3-58}$$

Por outro lado, $\sum_{\ell=0}^{n-1} c_{i\ell} = W(\underline{c}_i)$, onde $W(\underline{c}_i)$ representa o peso da palavra $\underline{c}_i$. A variável aleatória $X_i \triangleq \sum_{\ell=0}^{n-1} c_{i\ell}\, n_\ell$ é uma combinação linear de variáveis aleatórias gaussianas independentes, o que acarreta em X_i também ser uma variável gaussiana |9|, com distribuição de probabilidade $X_i \sim \mathcal{N}(0, W(\underline{c}_i)\frac{\eta_0}{2})$. Portanto,

$$P(A_i) = P(X_i > \frac{\sqrt{E}}{2} W(\underline{c}_i)) = \frac{1}{\sqrt{\Pi \eta_0}} \int_{\frac{\sqrt{E}}{2} W(\underline{c}_i)}^{+\infty} e^{-x^2/\eta_0 W(\underline{c}_i)} dx \tag{3-59}$$

ou seja,

$$P(A_i) = Q(\sqrt{E\, W(\underline{c}_i)/2\eta_0})$$

$$= Q(\sqrt{R\gamma_b\, W(\underline{c}_i)/2}) \tag{3-60}$$

A cota (3-58) é apertada para alta relação sinal/ruído |19|, de modo que é possível escrever

$$AB_{\gamma b\to\infty}(P_{bloco}) \approx \sum_{i=1}^{2^k-1} P(A_i) \qquad (3\text{-}61)$$

Contudo, somente palavras código com peso d contribuem substancialmente no somatório em (3-61), visto que para valores suficientemente grandes de x, $Q(x) \approx \frac{1}{x\sqrt{2\Pi}} \exp(-x^2/2)$. Logo, o comportamento assintótico para a probabilidade de se decodificar incorretamente uma palavra quando a regra ótima é utilizada em um canal gaussiano com alta relação sinal/ruído é expresso por

$$AB_{\gamma b\to\infty}(P_{bloco}) \approx N(d)\ Q(\sqrt{Rd\gamma_b/2})$$

Considerando pulsos polares de amplitude $+\sqrt{E}$ volts e $-\sqrt{E}$ volts, o resultado obtido coincide exatamente com o comportamento assintótico determinado para o algorítmo de Hartmann-Rudolph na seção 3.1.4.

CAPÍTULO IV

SIMULAÇÃO EM COMPUTADOR

Neste capítulo são apresentados os resultados de simulações realizadas em computador digital (DEC-10 SYSTEM) para análise do desempenho dos dois algorítmos descritos no capítulo anterior. A necessidade desta simulação advém da complexidade* do procedimento analítico da variação da probabilidade de erro (por símbolo ou bloco) em função da relação sinal/ruído, e da degradação que resulta quando decisão suave é utilizada quantizando em Q regiões os símbolos recebidos do canal. O desempenho de vários códigos é analisado, e em particular, os códigos de Hamming (7,4,3), (15,11,3) e (31,26,3) são considerados. O intuito é fazer um levantamento das curvas de probabilidade de erro versus relação simal/ruído para cada código de bloco analisado, em um dado canal sem memória. Isto torna possível, por exemplo, que seja feita uma visualização da melhoria provida pelo uso de decisão suave com relação a decisão abrupta e com relação ao número de regiões de quantização utilizado.

Os programas foram escritos na linguagem FORTRAN 10 e se encontram listados no apêndice A.

*Veja apêndice B.

4.1 - ANÁLISE DO DESEMPENHO DO ALGORÍTMO DE HARTMANN-RUDOLPH

4.1.1 - Considerações Gerais

O desempenho do algorítmo de Hartmann-Rudolph será analisado para vários códigos de bloco lineares, quando os sinais transmitidos são pulsos liga-desliga sujeitos a ação de um ruído aditivo gaussiano. Em particular, resultados para os códigos de Hamming são apresentados.

Na simulação, primeiro é especificado o código linear binário $C(n,k,d)$ fornecendo sua matriz de verificação de paridade $[H]$, ou o polinômio gerador $g(x)$ no caso de códigos cíclicos. A seguir é fixado o número de regiões de quantização a ser utilizado na simulação, usualmente uma potência de dois. A análise é realizada considerando 2,4,8 ou até 16 regiões de quantização.

A simulação é feita transmitindo-se a palavra toda zero em um canal com ruído aditivo gaussiano, gerado de acordo com o método polar |22|. Não há nenhuma perda de generalidade em se assumir que a palavra transmitida foi a palavra toda nula |19|.

Para os códigos de Hamming analisados, a opção de utilizar a palavra transmitida como sendo a palavra toda um, forneceu praticamente os mesmos resultados obtidos quando a palavra transmitida foi a toda nula, mostrando que o ruído gerado apresenta simetria. Os resultados para o código (7,4,3) também foram verificados considerando os bits do bloco de mensagem gerados de acordo com uma sequência pseudo-aleatória |11|.

As probabilidades de erro por símbolo e por bloco são então estimadas para cada valor de relação sinal/ruído (em decibeis) transmitindo um número suficientemente grande de palavras código através do canal e utilizando a regra ótima de decodificação no receptor. A estimativa é feita utilizando a frequência relativa de

ocorrência de erros na saída do decodificador.

A utilização da versão fraca da lei dos grandes números de acordo com o teorema de BERNOULLI | 9 | assegura que os resultados obtidos convergem para os valores reais das probabilidades, pois se n_ℓ/n é a frequência relativa da ocorrência de erros, então

$$P\left[\left|\frac{n_\ell}{n} - p\right| \geq \varepsilon\right] \leq \frac{1}{4n\varepsilon^2} \text{ para todo } \varepsilon > 0. \qquad (4\text{-}1)$$

Deve ser notado que o número de palavras a serem transmitidas pela fonte deve ser aumentado à medida que a relação sinal/ruído cresce. Tal aumento de relação sinal/ruído acarreta uma diminuição no número de erros provocados pelo canal e consequentemente a interpretação frequentista utilizada para o cálculo das probabilidades de erro poderá não mais ser válida. Pode ser facilmente demonstrado |33| que para uma variável aleatória com distribuição binominal $\mathcal{B}(n,p)$, um estimador não enviezado para p é a frequência relativa $\hat{p} = k/n$, o qual resulta em um espalhamento relativo em torno do valor esperado da estimativa dado por

$$\frac{\hat{\sigma}}{E\{\hat{p}\}} = \frac{1}{\sqrt{n}}\sqrt{\frac{q}{p}} \qquad (4\text{-}2)$$

Se p representa a probabilidade de erro por bit na saída do decodificador, e q a probabilidade do evento complementar, então para $p \ll 1$ segue-se $q \approx 1$, de modo que

$$n \approx \frac{1}{p\,|\hat{\sigma}/E\{\hat{p}\}|^2} \qquad (4\text{-}3)$$

Por exemplo, se a probabilidade de erro é da ordem de 10^{-3} e é desejado um espalhamento relativo de 10%, então a simulação deve ser realizada para n da ordem de 10^5 bits transmitidos. Desta forma, para SNR = 4dB é suficiente que o número de bits

transmitidos seja da ordem de mil bits, enquanto que para SNR = 12 dB é necessário que este número seja cerca de 100 mil bits, para assegurar a validade dos resultados encontrados.

Como a palavra decodificada pelo algorítmo não é necessariamente uma palavra código, surgiu a idéia de escolher como estimativa da palavra transmitida a palavra código mais próxima em termos de distância de Hamming da palavra decodificada. Contudo, os resultados obtidos como o emprego de tal procedimento degradaram sensivelmente a probabilidade de erro por símbolo, enquanto que a melhoria na probabilidade de erro por bloco foi praticamente nula.

O tempo de CPU requerido para realização das simulações é razoavelmente grande, como seria de se esperar, não chegando contudo, a tornar-se proibitivo, exceto para SNR's muito elevadas. Contudo, para altas relações sinal/ruído, o comportamento assintótico do algorítmo é conhecido.

De posse dos resultados obtidos, foram traçadas as curvas de probabilidade de erro versus relação sinal/ruído mostradas a seguir, nas páginas 108 a 124.

4.1.2 - Curvas de Desempenho

As curvas de desempenho para o algorítmo de Hartmann - Rudolph obtidas através de simulação em computador digital estão apresentadas nesta seção. É admitido um canal perturbado por um ruído branco gaussiano aditivo em todos os casos. O número de regiões de quantização é abreviado por NQ e PBIT denota a probabilidade de erro por símbolo, enquanto que PBLOCO denota a probabilidade de erro por palavra.

Figura 4.1 - SIMULAÇÃO PARA O CÓDIGO(7,4,3)

Curvas PBIT x SNR

1 - Decisão abrupta NQ = 2

2 - Decisão suave NQ = ∞

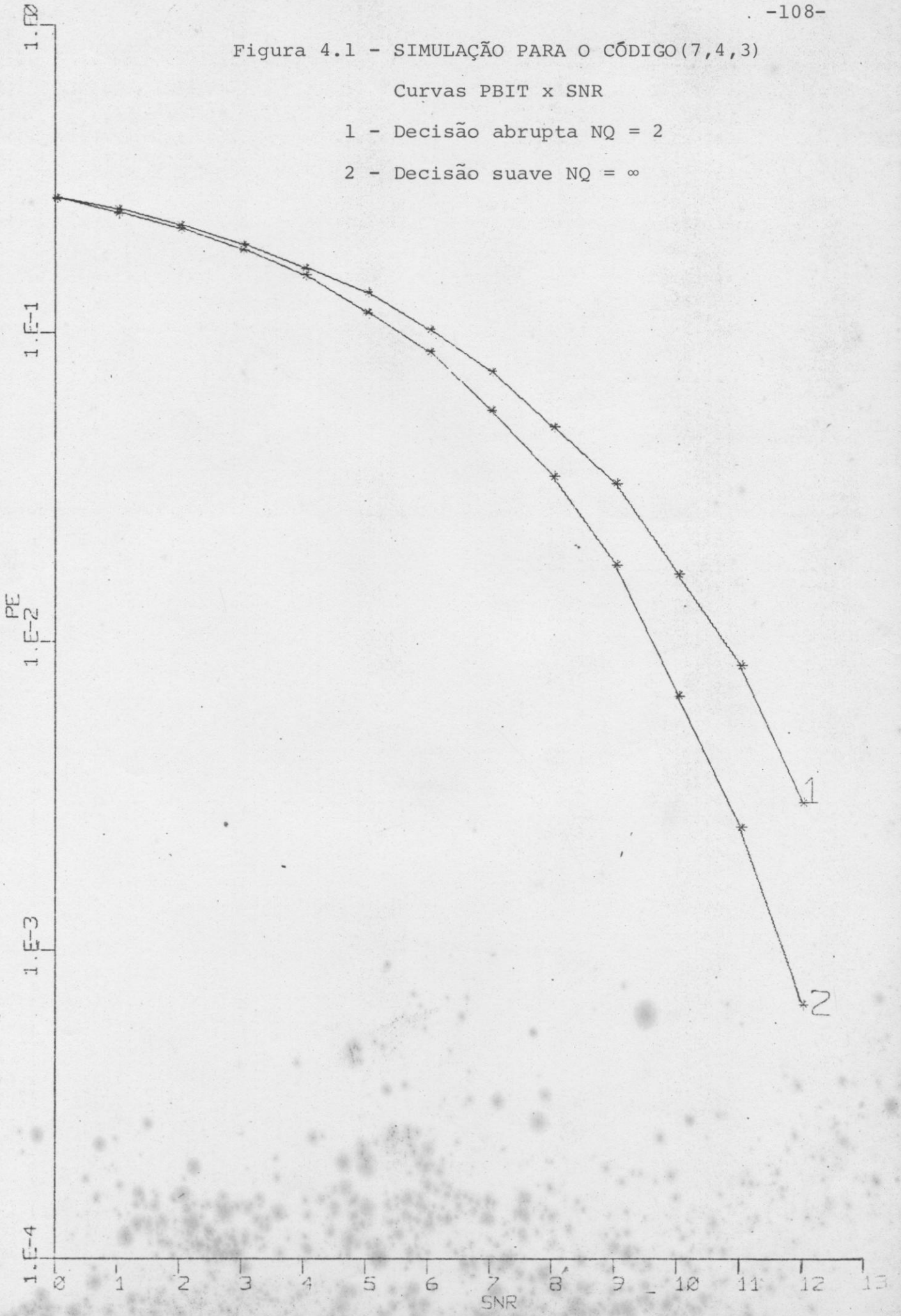


Figura 4.2 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

Curvas PBLOCO x SNR

1 - Decisão abrupta NQ = 2

2 - Decisão suave NQ = ∞

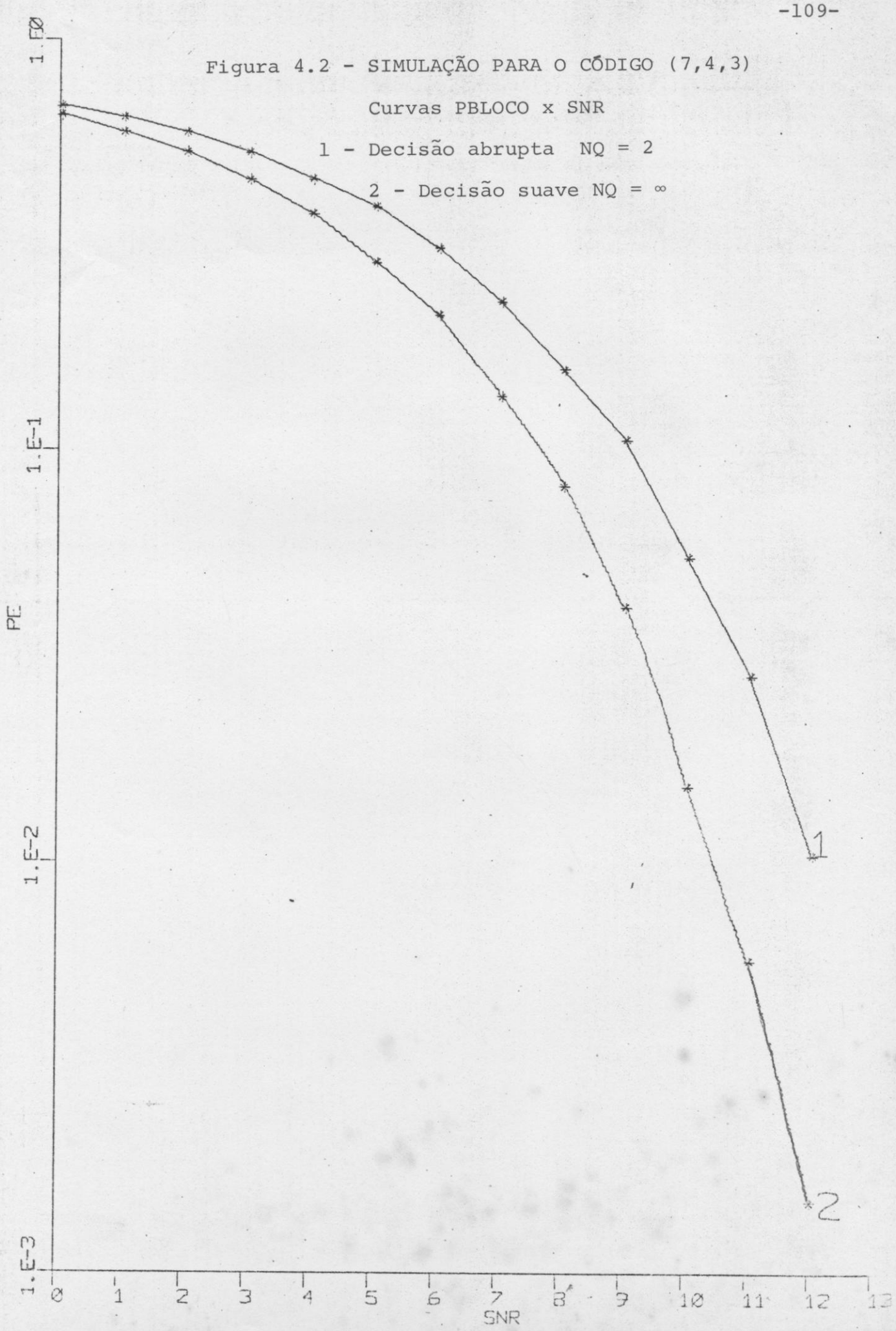


Figura 4.3 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

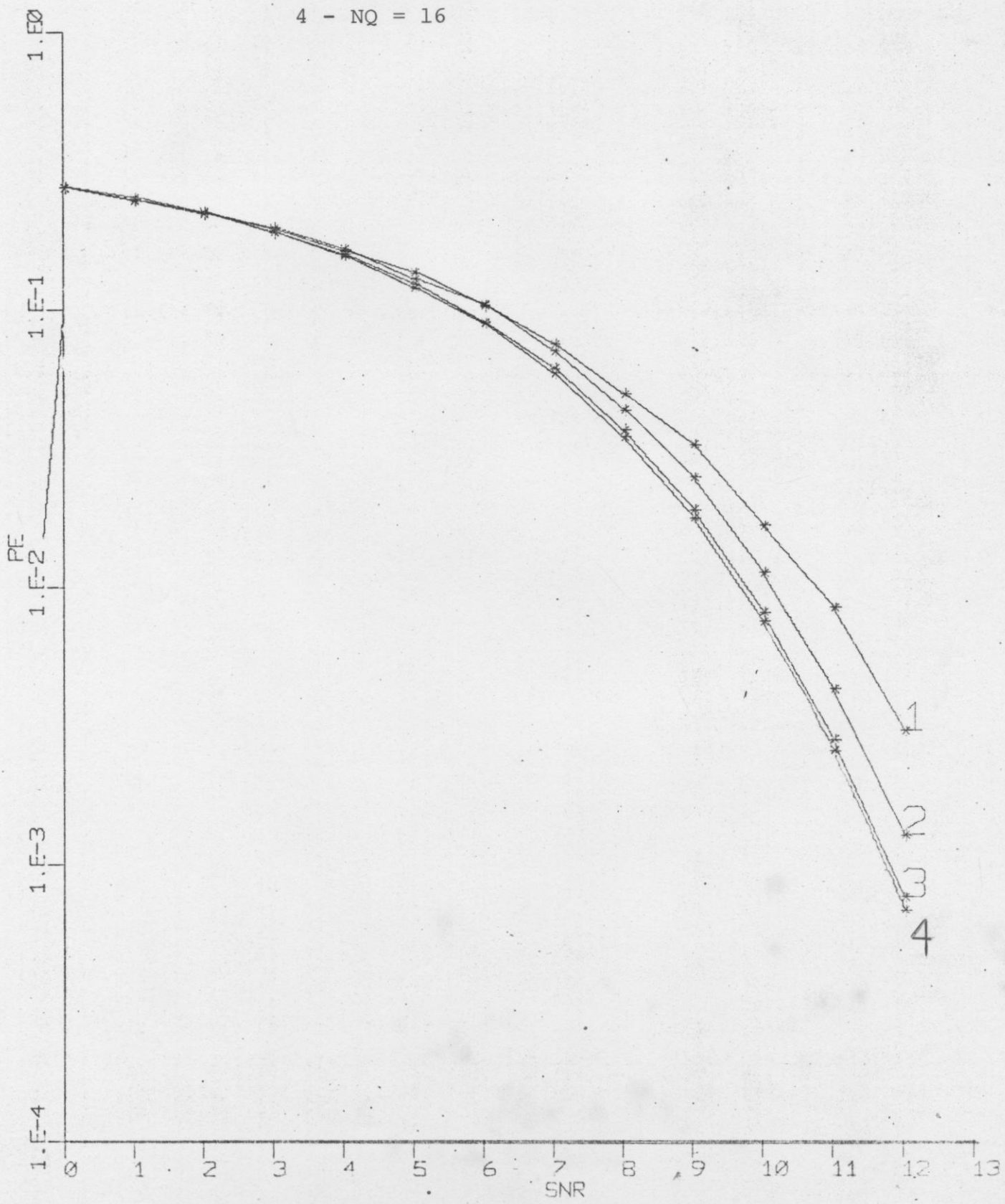
Curvas PBIT x SNR

1 - NQ = 2

2 - NQ = 4

3 - NQ = 8

4 - NQ = 16



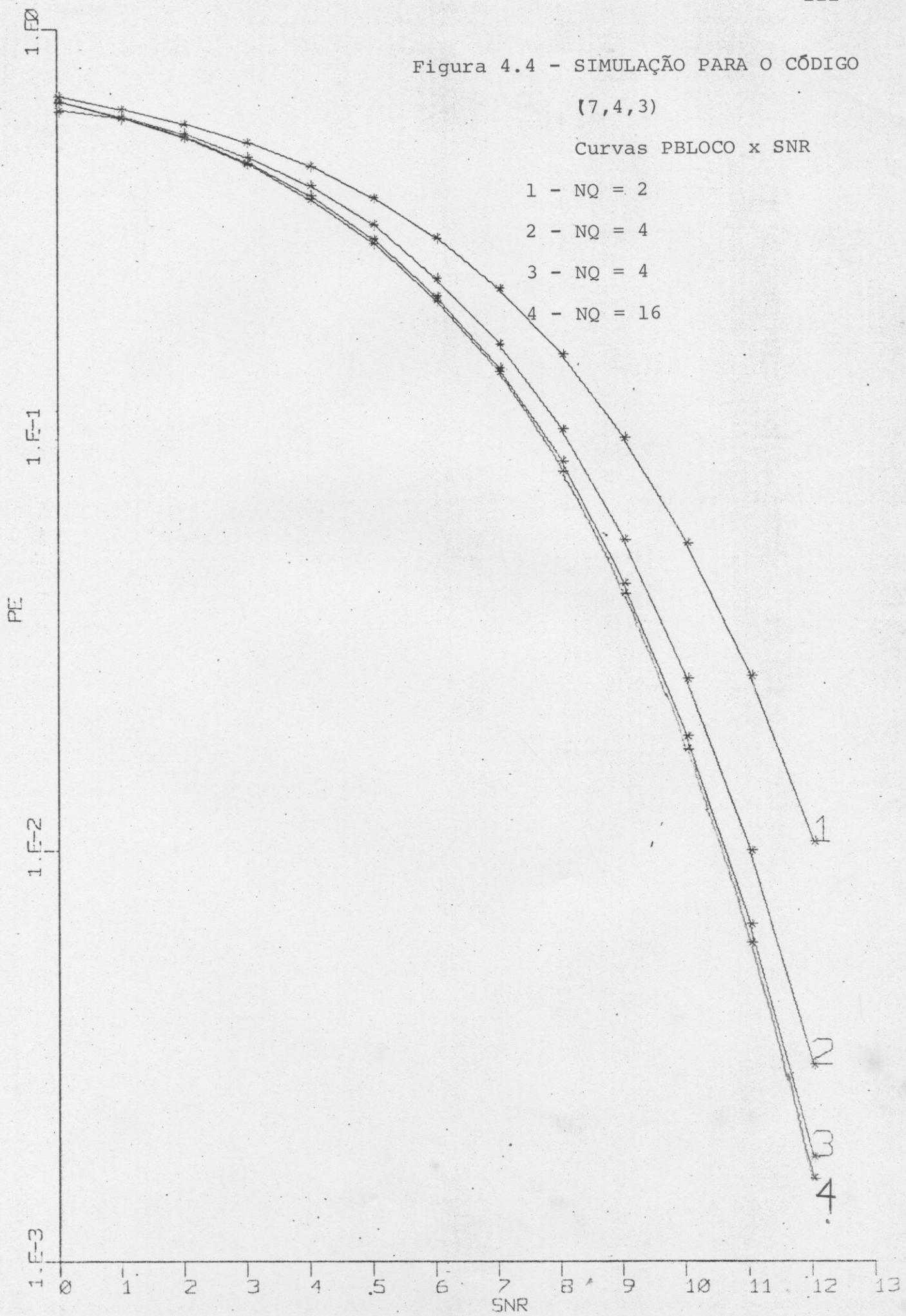


Figura 4.4 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

Curvas PBLOCO x SNR

1 - NQ = 2

2 - NQ = 4

3 - NQ = 4

4 - NQ = 16

Figura 4.5 - SIMULAÇÃO PARA O CÓDIGO (15,11,3)

Curvas PBIT x SNR

1 - NQ = 4

2 - NQ = 8

3 - NQ = 16

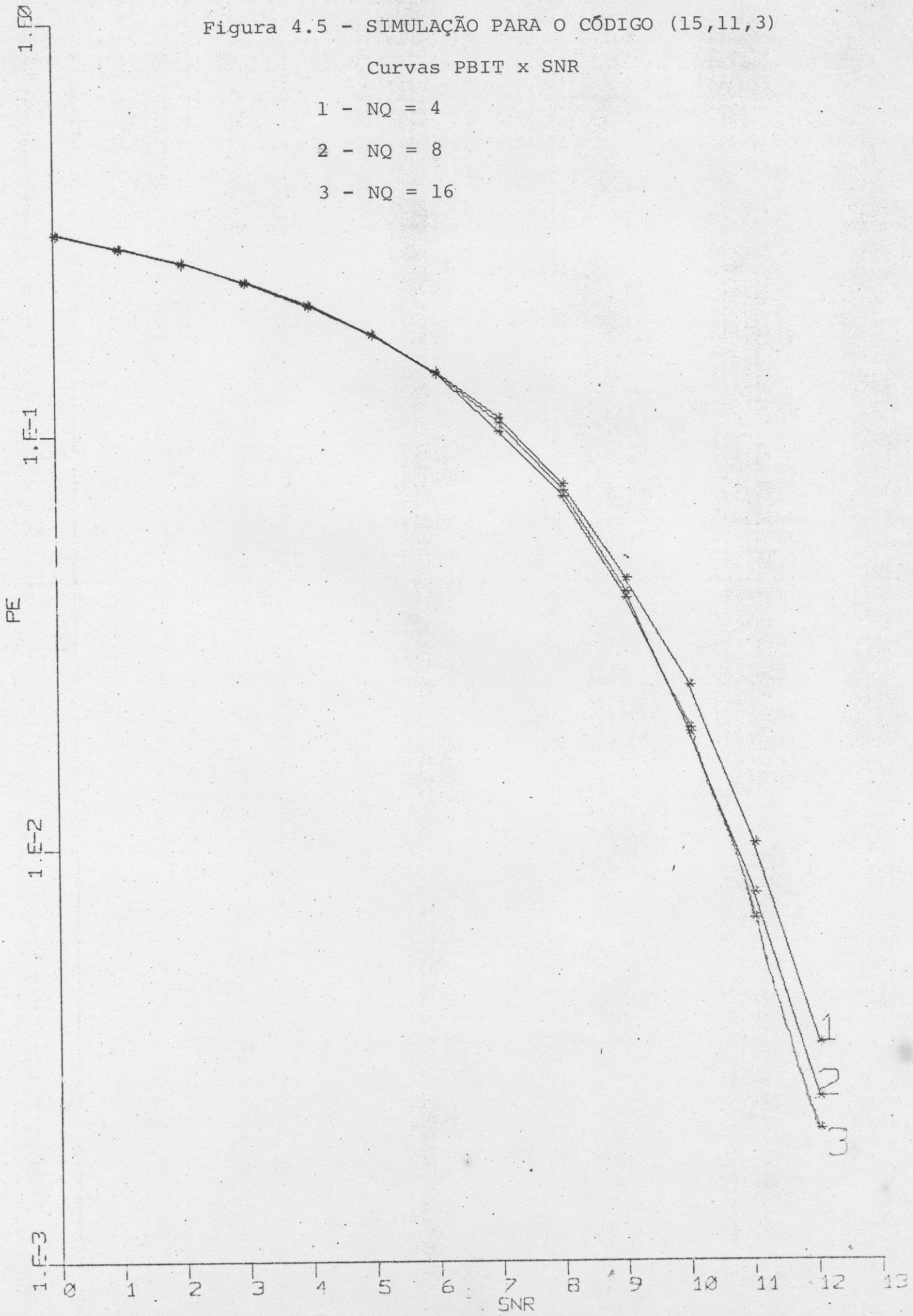


Figura 4.6 - SIMULAÇÃO PARA O CÓDIGO (15,11,3)

Curvas PBLOCO x SNR

1 - NQ = 2

2 - NQ = 4

3 - NQ = 8

4 - NQ = 16

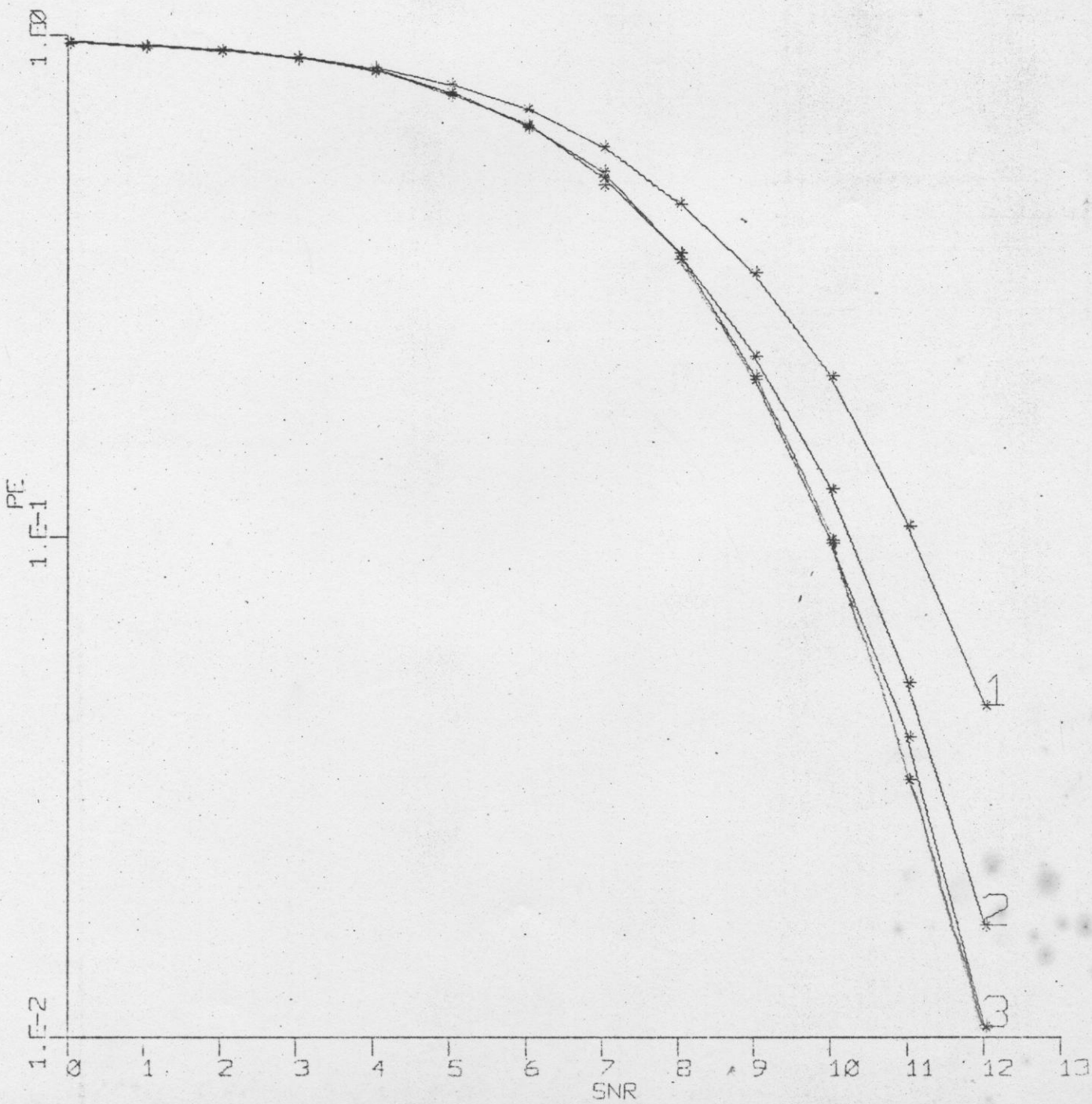


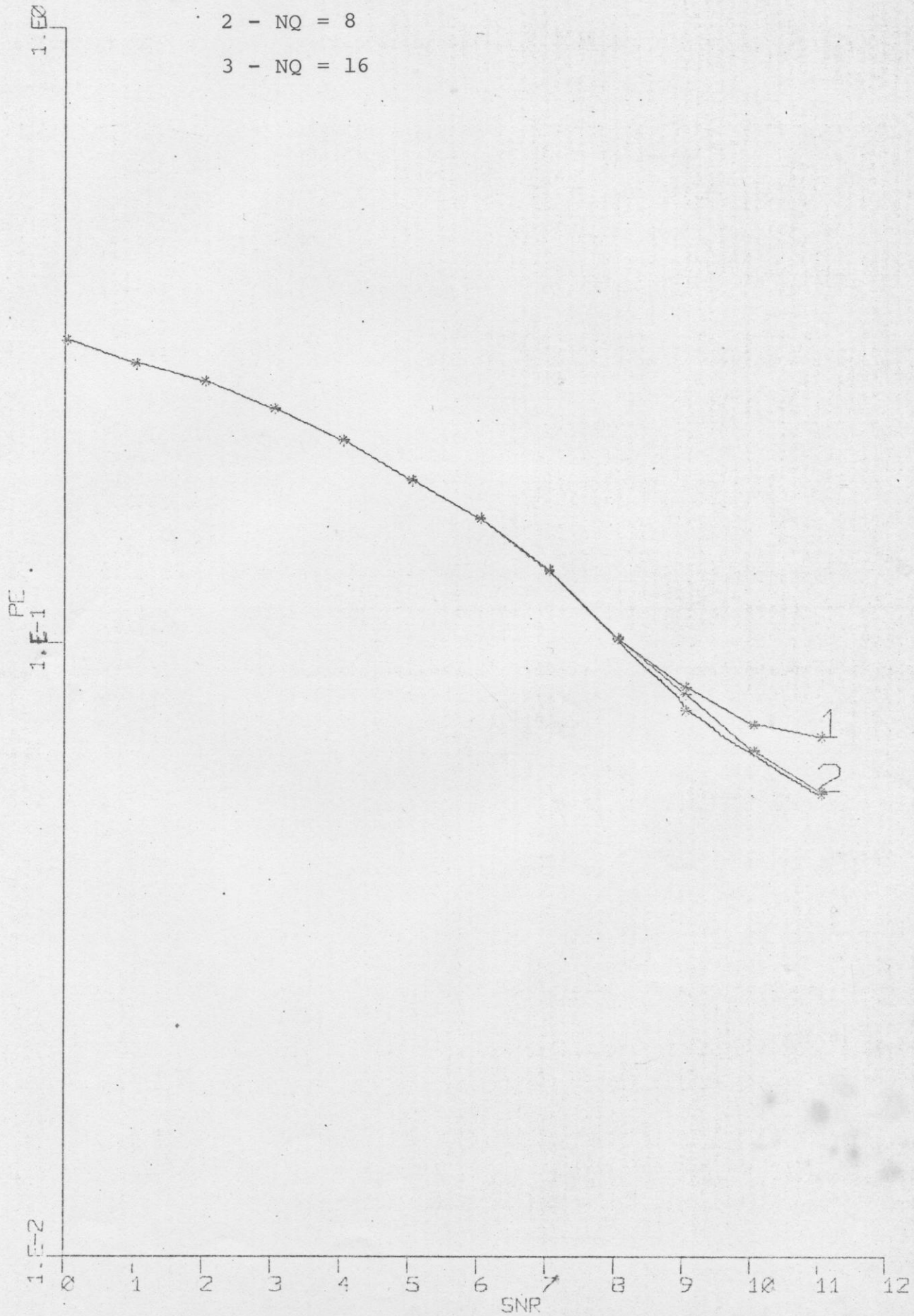
Figura 4.7 - SIMULAÇÃO PARA O CÓDIGO (31,26,3)

Curvas PBIT x SNR

1 - NQ = 4

2 - NQ = 8

3 - NQ = 16



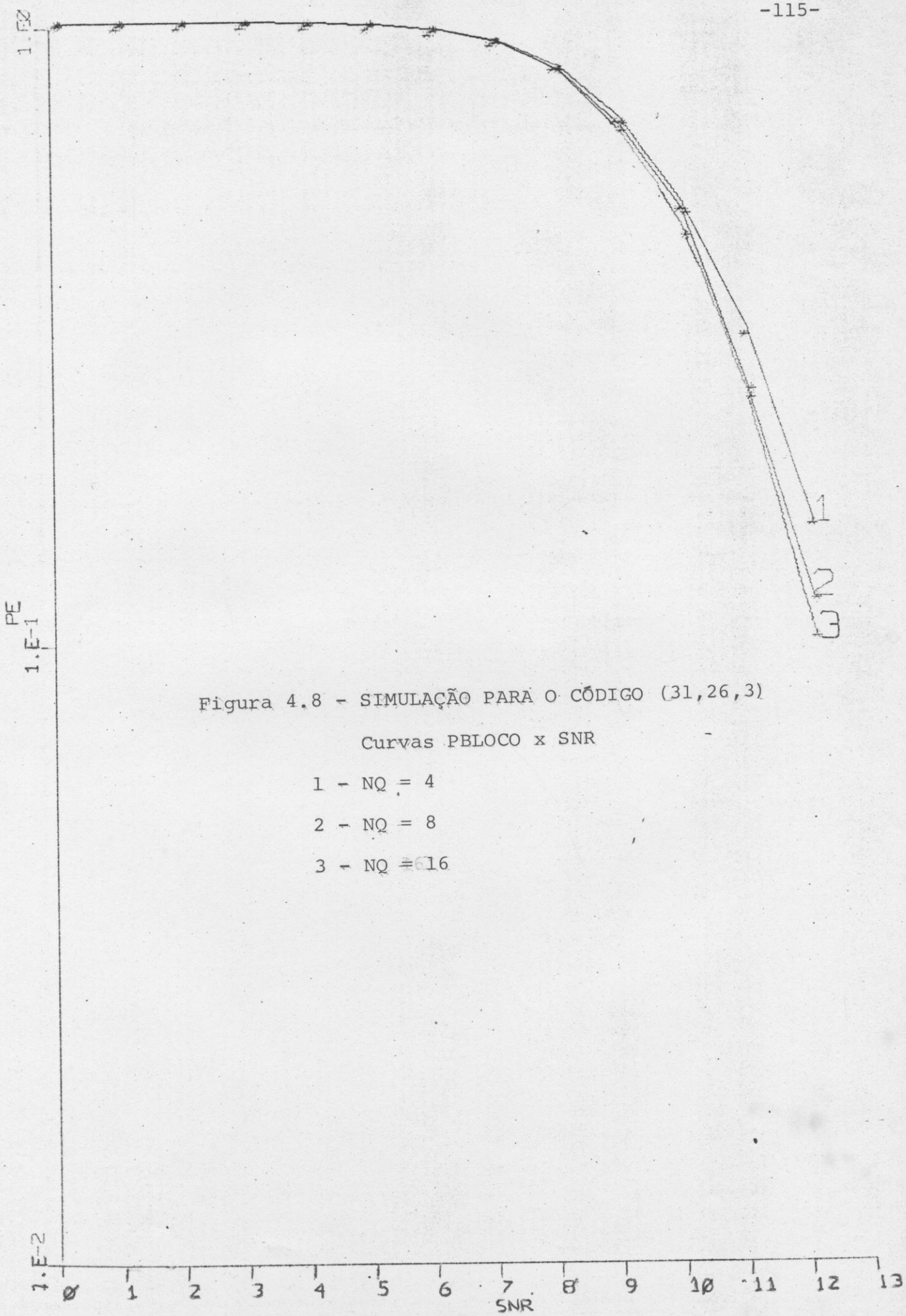


Figura 4.8 - SIMULAÇÃO PARA O CÓDIGO (31,26,3)

Curvas PBLOCO x SNR

1 - NQ = 4

2 - NQ = 8

3 - NQ = 16

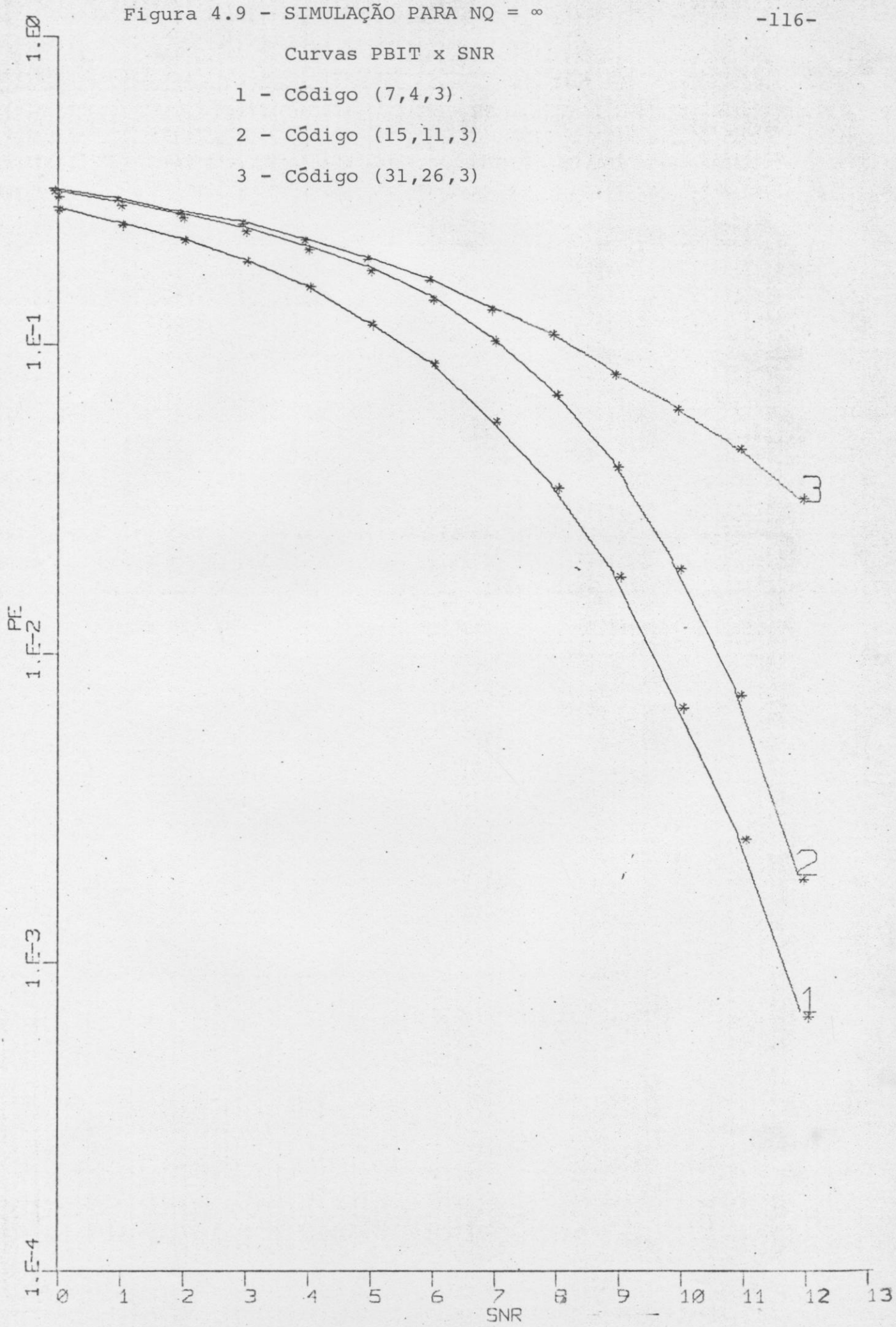
Figura 4.9 - SIMULAÇÃO PARA NQ = ∞

Curvas PBIT x SNR

1 - Código (7,4,3)

2 - Código (15,11,3)

3 - Código (31,26,3)



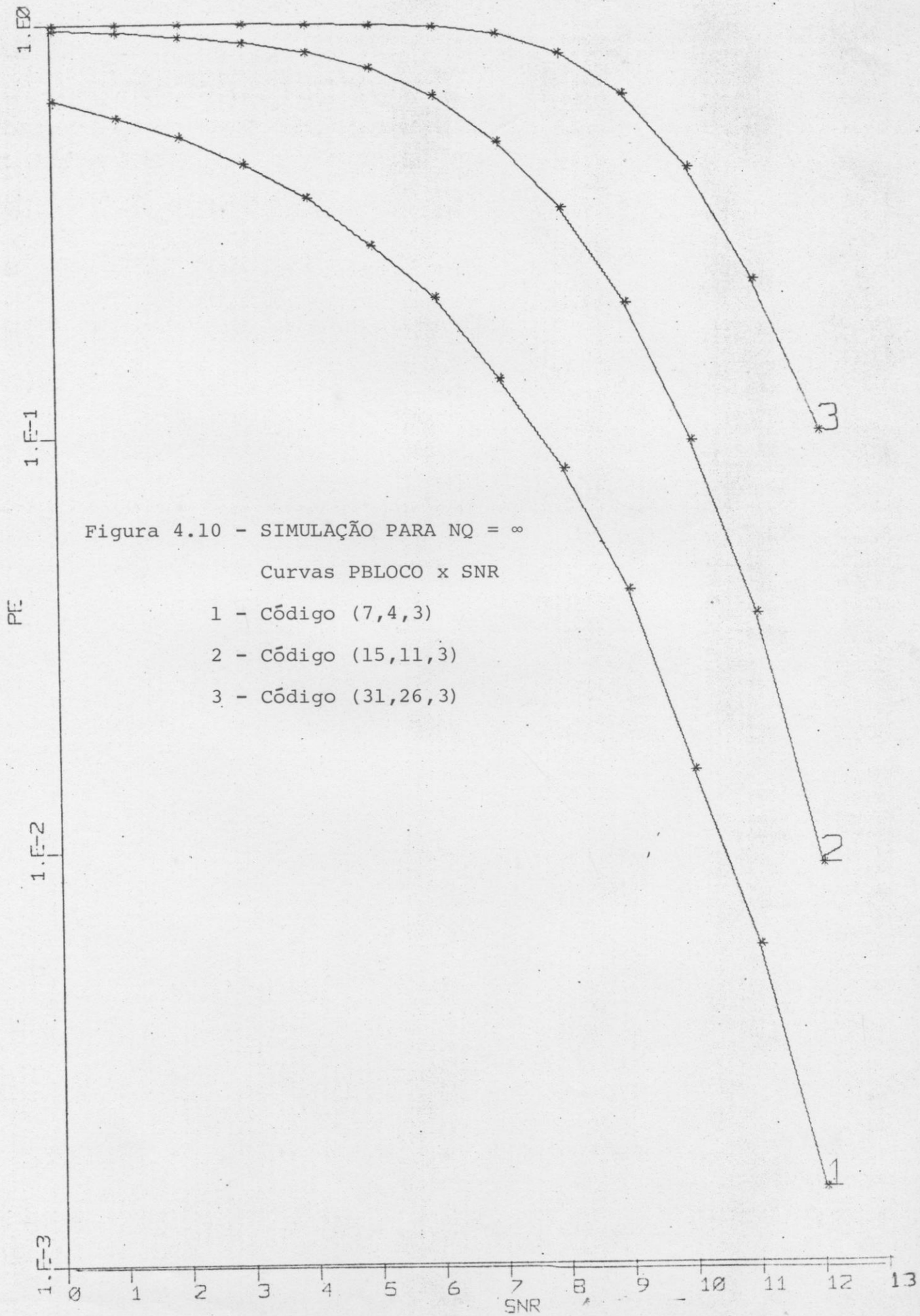


Figura 4.10 - SIMULAÇÃO PARA NQ = ∞

Curvas PBLOCO x SNR

1 - Código (7,4,3)

2 - Código (15,11,3)

3 - Código (31,26,3)

Figura 4.11 - SIMULAÇÃO PARA NQ = 2

Curvas PBLOCO x SNR

1 - Código (7,4,3)

2 - Código (15,11,3)

3 - Código (31,26,3)

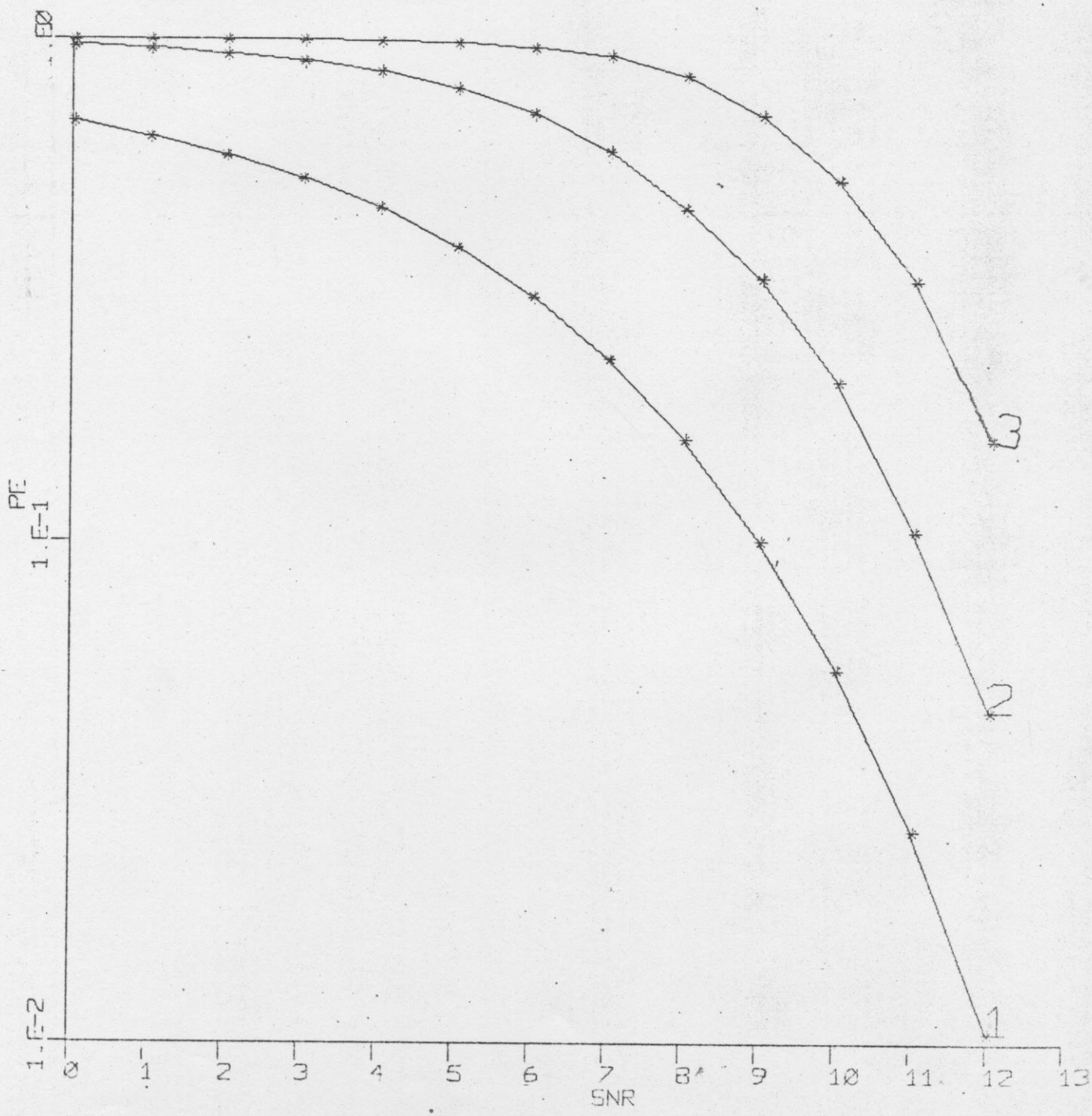


Figura 4.12 - SIMULAÇÃO PARA NQ = 4

Curvas PBIT x SNR

1 - Código (7,4,3)

2 - Código (15,11,3)

3 - Código (31,26,3)

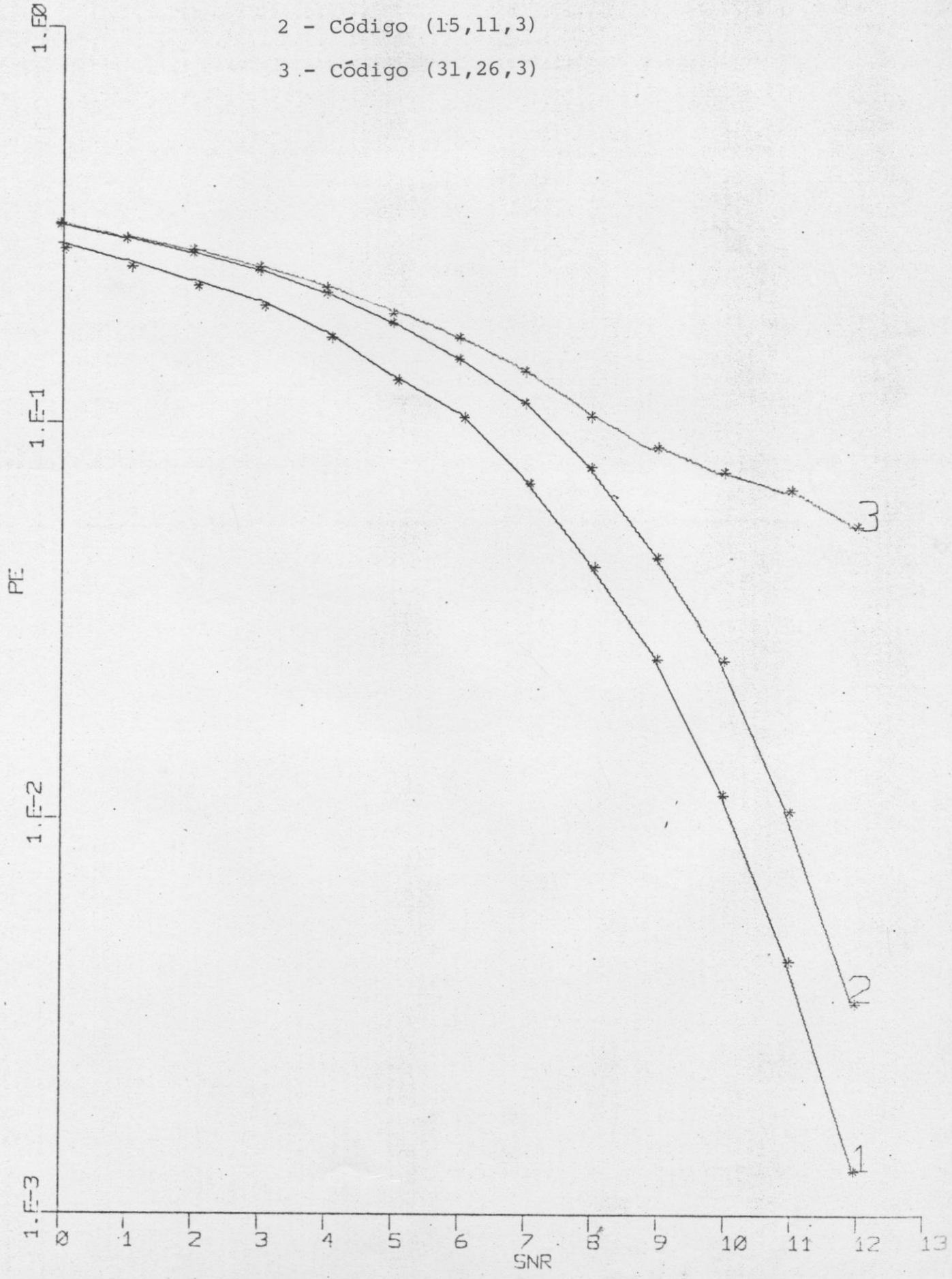


Figura 4.13 - SIMULAÇÃO PARA NQ = 4

Curvas PBLOCO x SNR

1 - Código (7,4,3)

2 - Código (15,11,3)

3 - Código (31,26,3)

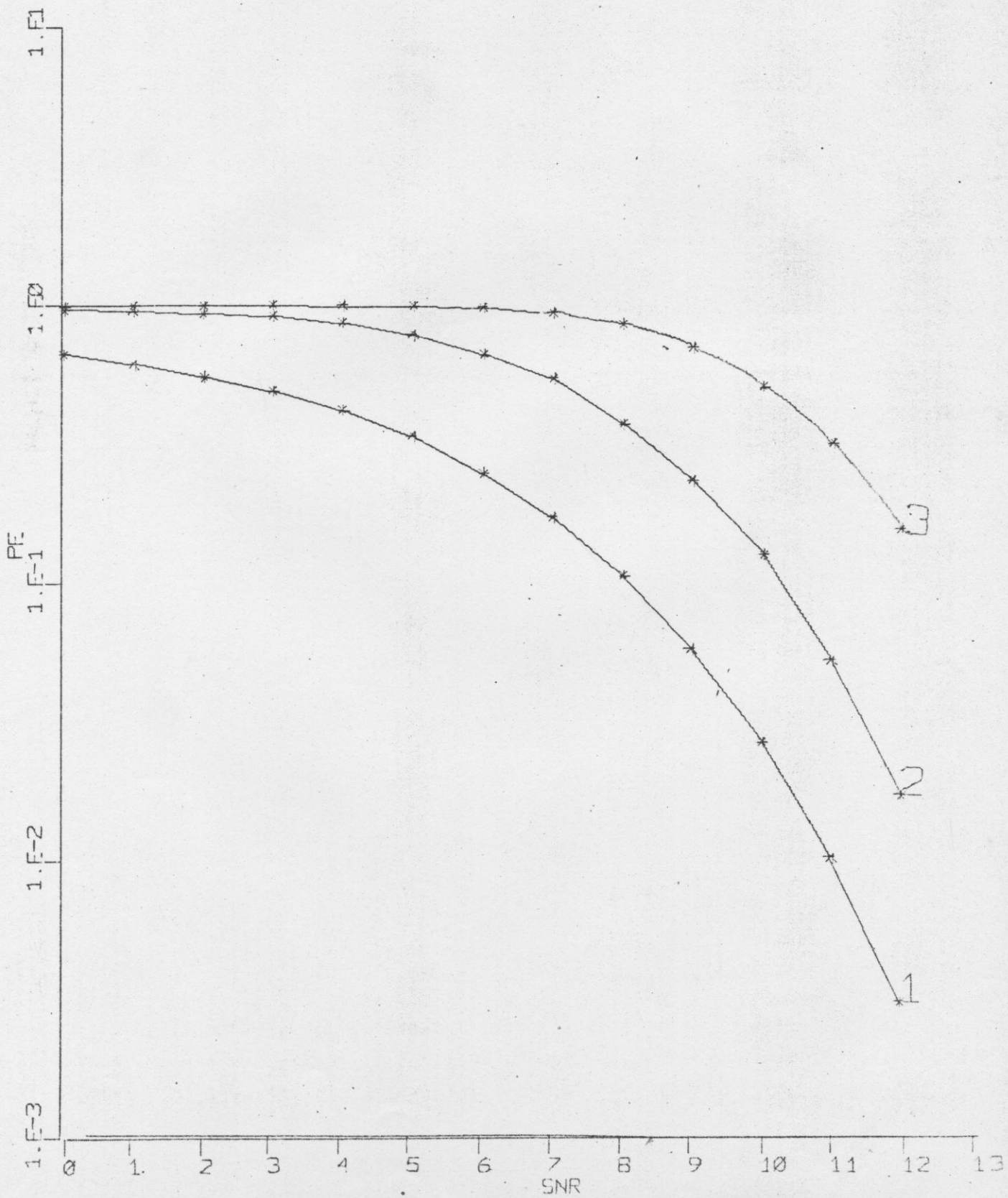


Figura 4.14 - SIMULAÇÃO PARA NQ = 8

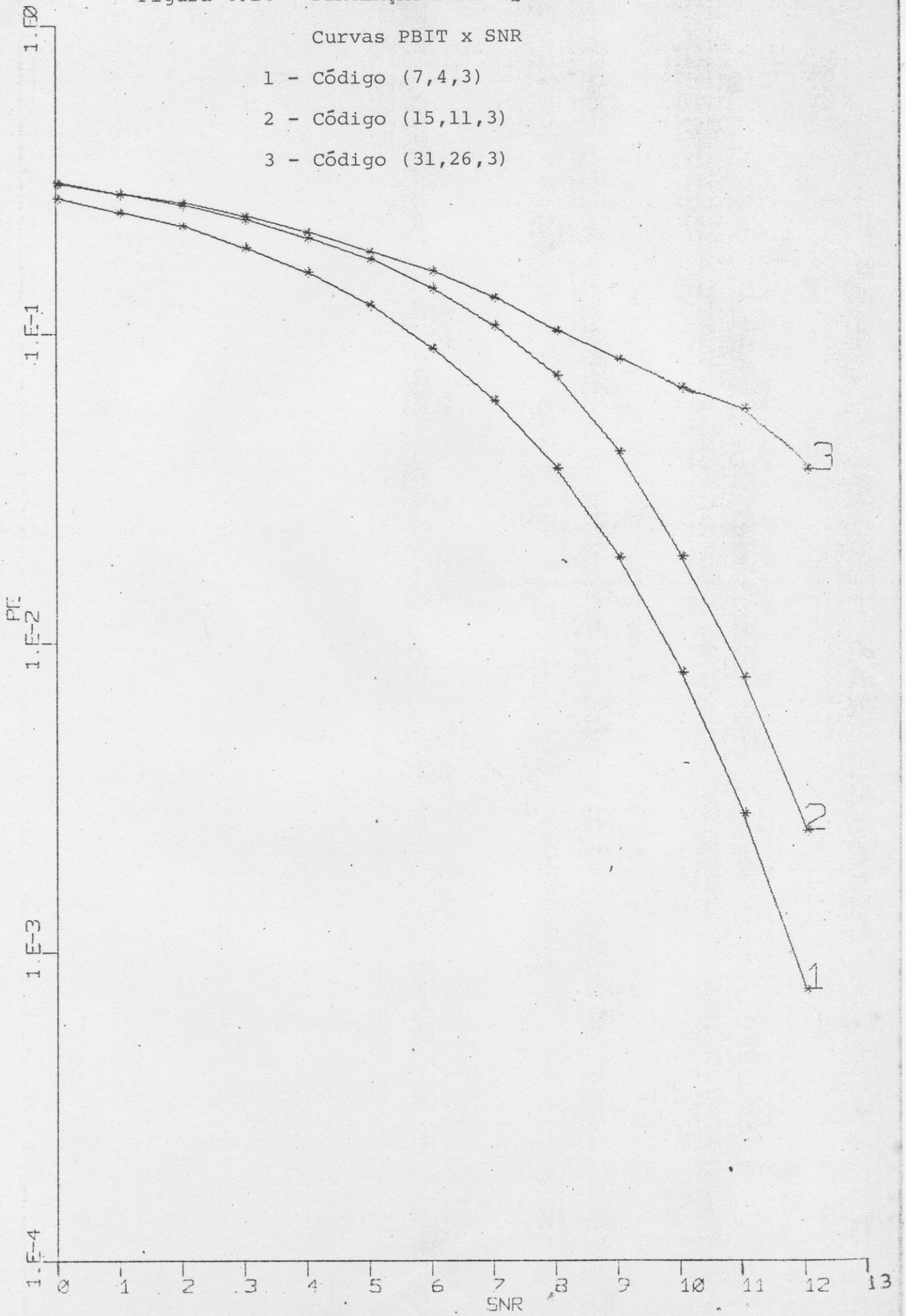


Figura 4.15 - SIMULAÇÃO PARA NQ = 8

Curvas PBLOCO x SNR

1 - Código (7,4,3)

2 - Código (15,11,3)

3 - Código (31,26,3)

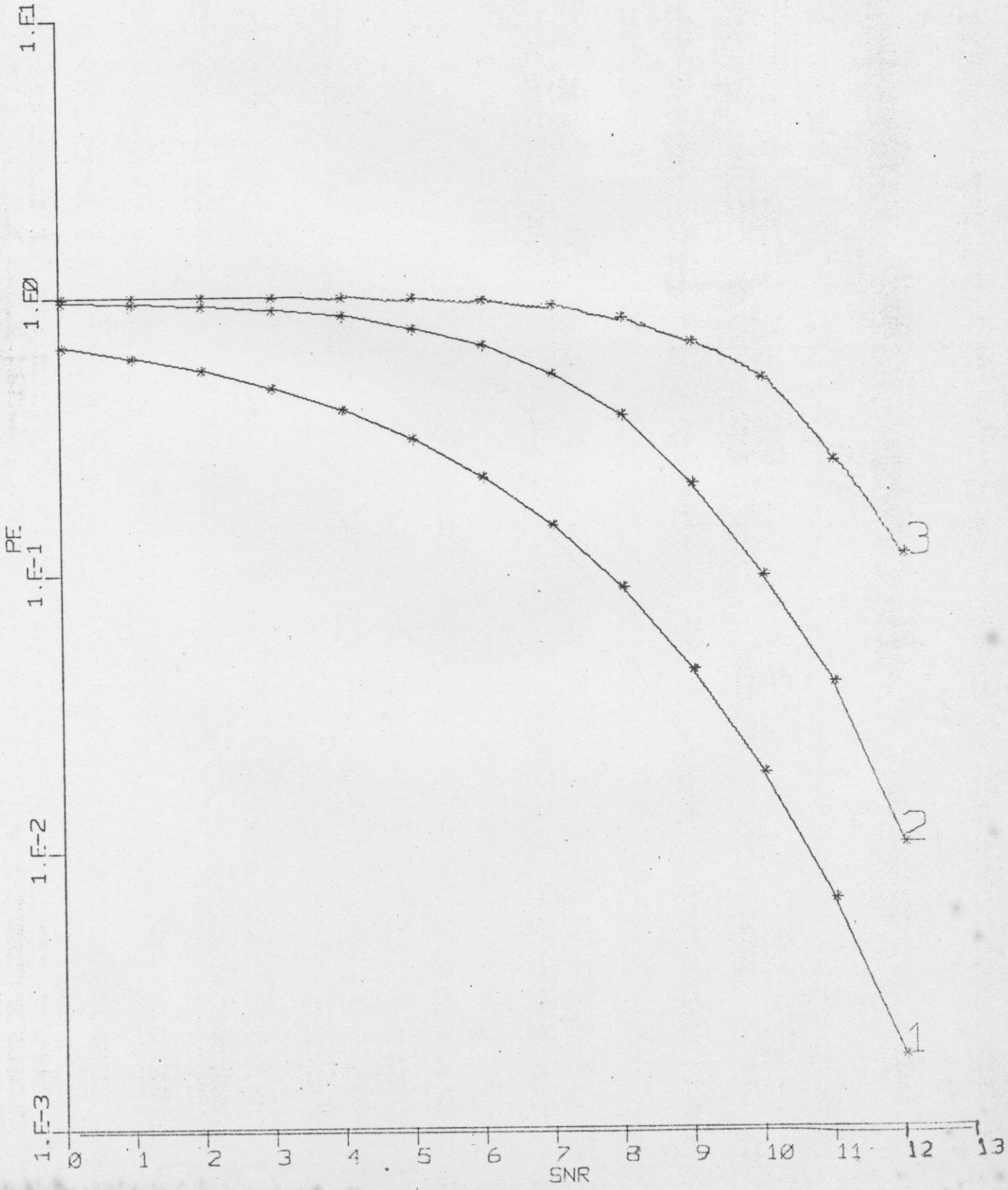


Figura 4.16 - SIMULAÇÃO PARA NQ = 16

Curvas PBIT x SNR

1 - Código (7,4,3)

2 - Código (15,11,3)

3 - Código (31,26,3)

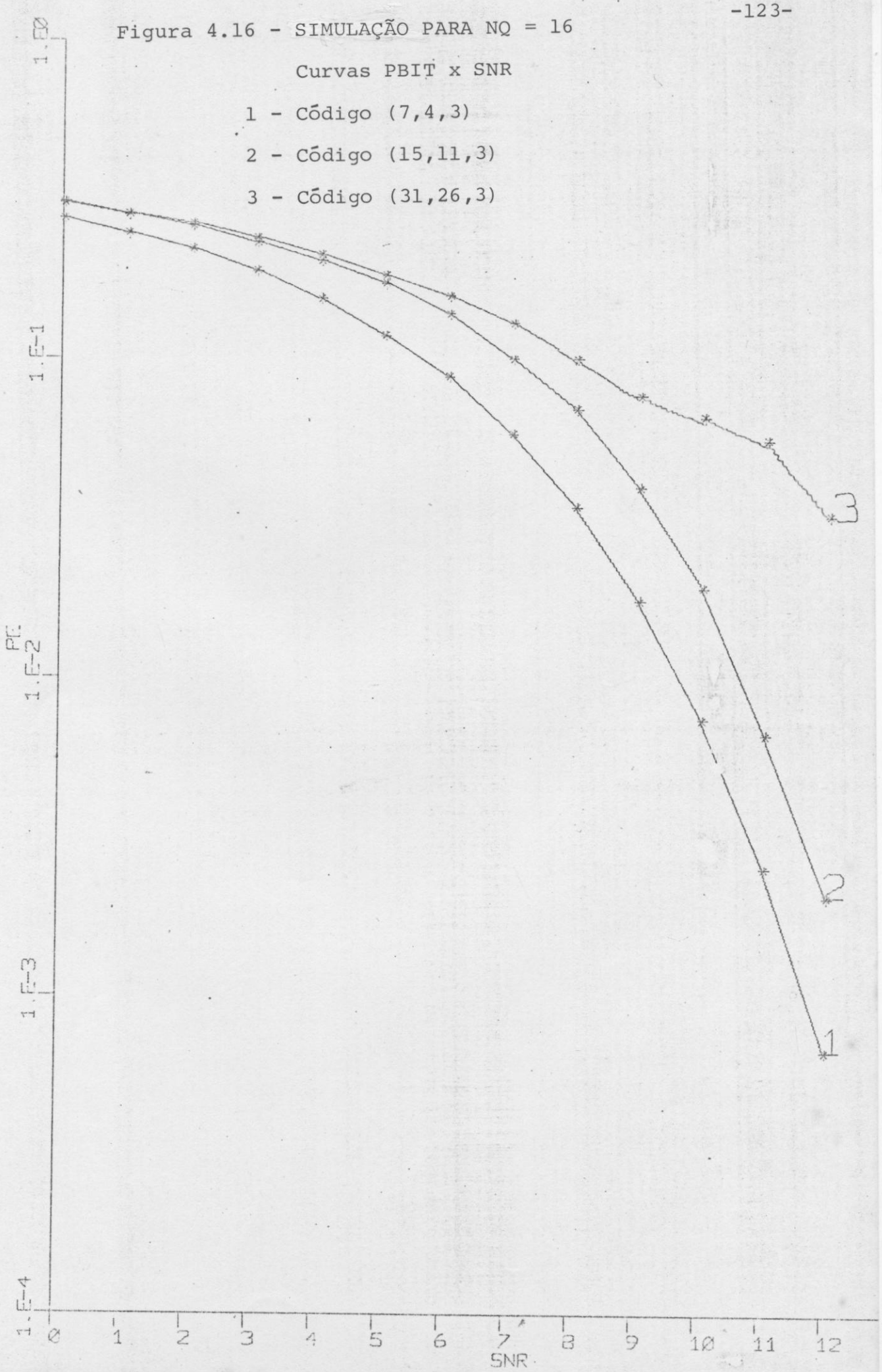


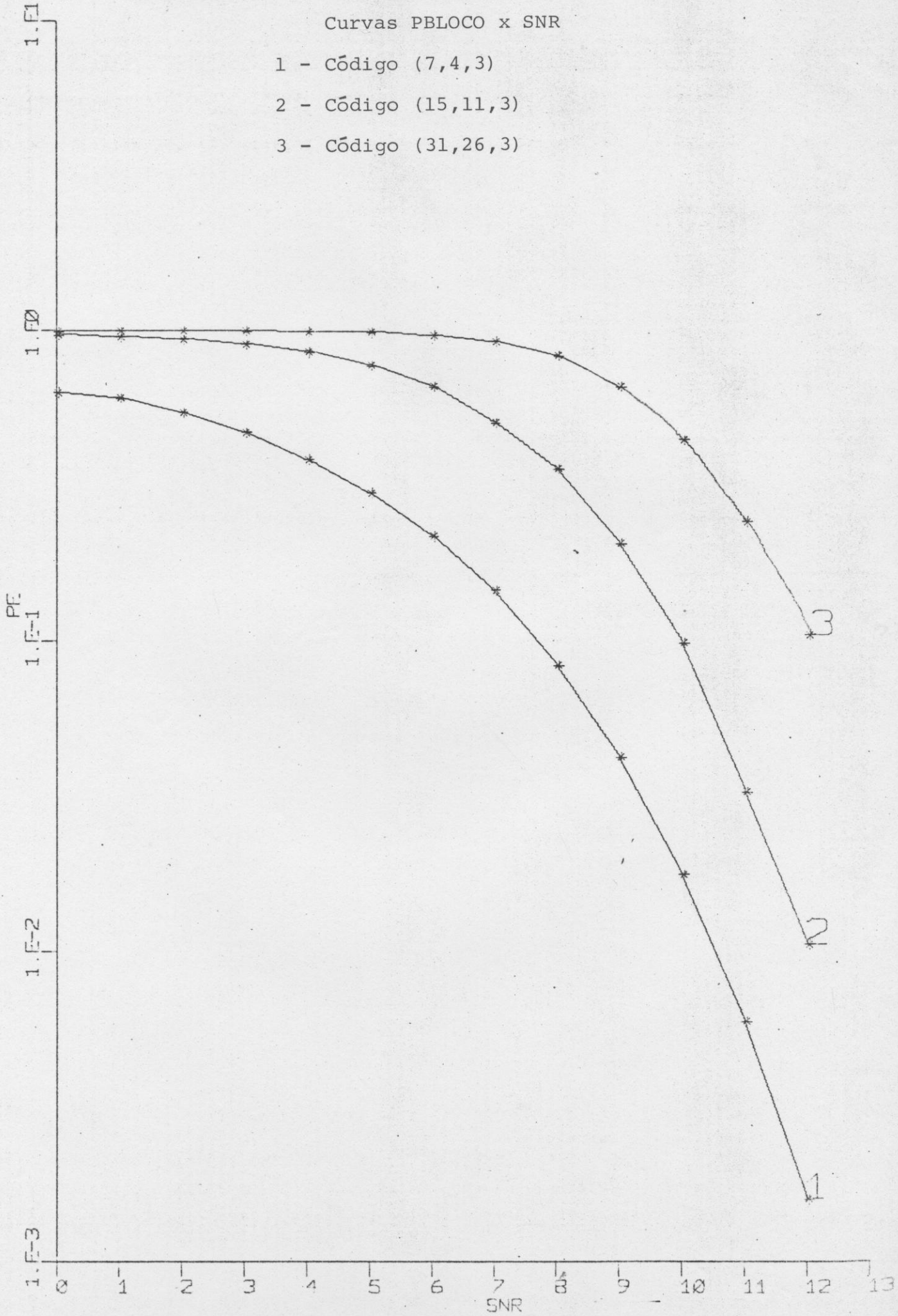
Figura 4.17 - SIMULAÇÃO PARA NQ = 16

Curvas PBLOCO x SNR

1 - Código (7,4,3)

2 - Código (15,11,3)

3 - Código (31,26,3)



4.2 - DESEMPENHO DA DECODIFICAÇÃO UTILIZANDO MÁXIMA PROBABILIDADE A POSTERIORI

4.2.1 - Considerações Gerais

Uma avaliação será apresentada do desempenho do segundo algorítmo ótimo descrito no capítulo anterior(sec.3.2)para vários ' códigos de bloco lineares, supondo sinais transmitidos em pulsos de RF liga-desliga através de um canal ruidoso.

O algorítmo fará uso da treliça associada ao código para simplificar as operações de decodificação. Inicialmente um código de bloco linear em GF(q) é especificado através de (n,k,d) e da sua matriz de verificação de paridade. A treliça associada ao código é então gerada de acordo com um procedimento melhor esclarecido' na seção 4.2.2.

As palavras código transmitidas pela fonte são geradas aleatoriamente e de tal forma que possuam uma distribuição de probabilidade uniforme. Deste modo, a hipótese requerida para otimalidade do algorítmo é obedecida. O ruído na representação de banda estreita é novamente simulado utilizando o método polar |22|. Serão abordados apenas códigos lineares binários, muito embora a treliça' possa também ser gerada para códigos de bloco lineares multiníveis. As probabilidades de erro por palavra e por bit quando este algorítmo de decodificação é utilizado na recepção são determinadas para ' cada valor de relação sinal/ruído (em dB), transmitindo um número ' suficientemente grande de palavras código através do canal ruidoso. A estimativa utilizada para determinação destas probabilidades é novamente a frequência relativa da ocorrência de erros, de modo que as considerações sobre convergência dos resultados discutidas na seção anterior ainda permanecem válidas. Com relação ao tempo de CPU requerido, considerações similares as enunciadas na seção anterior' também são aplicáveis.

4.2.2 - Implementação Computacional da Treliça

Será agora definida uma matriz associada à treliça gerada por um código de bloco linear em GF(q), com o intuito de permitir uma implementação da treliça sob o ponto de vista computacional |7|.

Seja [M] uma matriz com $q^{(n-k)}$ linhas e n colunas, onde cada elemento desta matriz é um vetor de q posições, i.e.,

$$[M] = \begin{vmatrix} \underline{m}_{1,1} & \underline{m}_{1,2} & \cdots\cdots & \underline{m}_{1,n} \\ \underline{m}_{2,1} & \underline{m}_{2,2} & \cdots\cdots & \underline{m}_{2,n} \\ \vdots & & & \\ \underline{m}_{q^{n-k},1} & \underline{m}_{q^{n-k},2} & \cdots\cdots & \underline{m}_{q^{n-k},n} \end{vmatrix} \qquad (4\text{-}4)$$

com $\underline{m}_{i,k} = (u_0, u_1, \ldots, u_{q-1})$ $\quad 1 \leq i \leq q^{n-k}$, $1 \leq k \leq n$

$u_\ell \in \{1, 2, \ldots, q^{n-k}\}$.

Portanto, o número de posições de memória necessário para definir uma treliça de um código linear (n,k,d) em GF(q) será de

$$n\, q^{n-k}.q = n\, q^{n-k+1} \text{ posições.} \qquad (4\text{-}5)$$

Cada vetor coluna de [M] está associado a uma profundidade $k > 0$ da treliça, e cada elemento $\underline{m}_{ik}$ deste vetor coluna está associado a um nó da mesma. Cada elemento $\underline{m}_{ik}$ por sua vez, possui sua coordenada u_ℓ igual a numeração correspondente ao nó k da profundidade anterior tal que $\underline{s}_i(k) = \underline{s}_j(k-1) \oplus \ell\, \underline{h}_k$. Deste

modo, cada vetor $\underline{m}_{ik}$ da profundidade k indica quais os nós e elementos $\alpha_j \in GF(q)$ que deram origem ao nó ao qual o vetor $\underline{m}_{ik}$ está associado. Fixados um dado nó NO e uma profundidade PROF, MATRIX(NO, PROF, ALFA), ALFA = 0,1,....,q-1 define uma q-upla correspondente a $\underline{m}_{NO,PROF}$. A primeira posição da q-upla significa que deve ser considerado que o nó em questão foi atingido através de $\alpha = 0$, enquanto a segunda posição significa que o ramo considerado se dirige para o nó em questão através de $\alpha = 1$, e assim por diante. Os valores indicados na q-upla fornecem os nós na profundidade anterior que se ligam com o nó em questão. É assumido que o valor zero indica que nenhuma ligação deve ser considerada. Como exemplo, o valor de MATRIX(3,2,0) fornece qual o nó na profundidade 1, o qual através de $\alpha = 0$ atinge o nó 3 na profundidade 2.

O procedimento utilizado para expurgar a treliça associada ao código de bloco considera inicialmente a construção de duas treliças auxiliares. A primeira delas, TRELLI 1, é gerada a partir da profundidade k = 0 até a profundidade k = n; enquanto que a segunda, TRELLI 2, é gerada iniciando na profundidade k = n até a profundidade k = 0. Os caminhos que coincidirem nas duas treliças fazem parte da treliça expurgada, em caso contrário o caminho é apagado. Este procedimento é melhor compreendido quando apresentado nos exemplos mostrados a seguir.

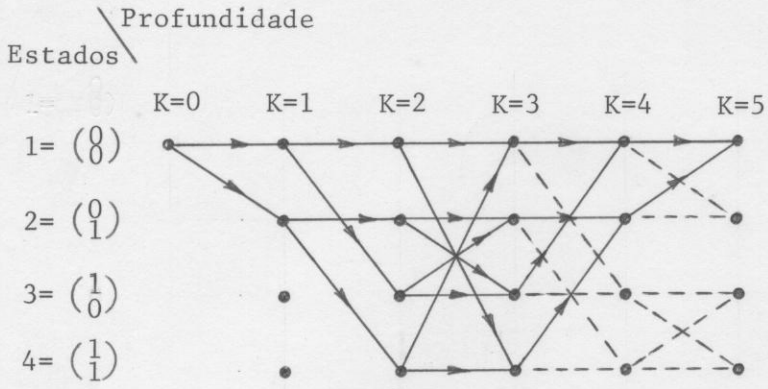
<u>Exemplo 4.1</u> - Caso binário.

Dado o código de bloco (5,3,2) com matriz de verificação de paridade [H]

$$[H] = \begin{vmatrix} 0 & 1 & 1 & \vdots & 1 & 0 \\ & & & \vdots & & \\ 1 & 0 & 1 & \vdots & 0 & 1 \end{vmatrix}$$

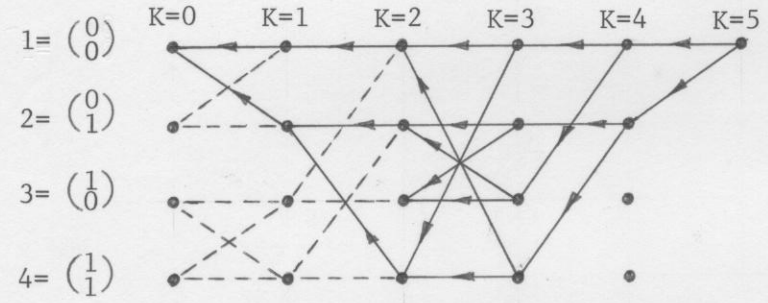
A matriz [M] que representa a treliça associada ao código é construída de acordo com os passos mostrados abaixo.

Tem-se n=5, k=3, n-k=2, d=2, q=2 e $q^{n-k} = 4$.



TRELLI 1

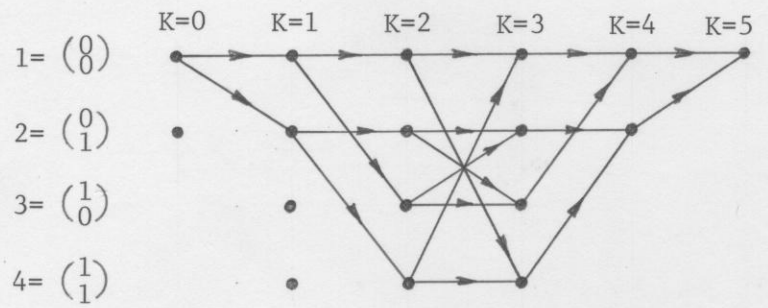
[1,0]	[1,0]	[1,4]	[1,3]	[1,2]
[0,1]	[2,0]	[2,3]	[2,4]	[2,1]
[0,0]	[0,1]	[3,2]	[3,1]	[3,4]
[0,0]	[0,2]	[4,1]	[4,2]	[4,3]



TRELLI 2

[1,2]	[1,3]	[1,4]	[1,3]	[1,2]
[2,1]	[2,4]	[2,3]	[2,4]	[0,0]
[3,4]	[3,1]	[3,2]	[0,0]	[0,0]
[4,3]	[4,2]	[4,1]	[0,0]	[0,0]

Finalmente a treliça expurgada associada ao código binário (5,3,2) é encontrada usando TRELLI 1 e TRELLI 2.



TRELLI

[1,0]	[1,0]	[1,4]	[1,3]	[1,2]
[0,1]	[2,0]	[2,3]	[2,4]	[0,0]
[0,0]	[0,1]	[3,2]	[0,0]	[0,0]
[0,0]	[0,2]	[4,1]	[0,0]	[0,0]

Exemplo 4.2 - Caso multinível.

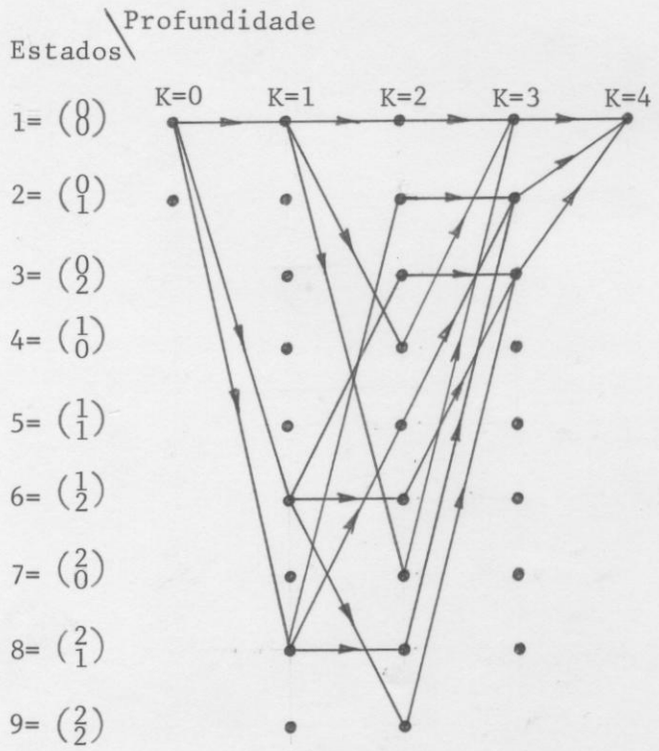
Como um segundo exemplo será considerado um código de bloco (4,2,3) definido em GF(3), com matriz de verificação de paridade dada por

$$[H] = \begin{vmatrix} 1 & 2 & \vdots & 1 & 0 \\ & & \vdots & & \\ 2 & 2 & \vdots & 0 & 1 \end{vmatrix}$$

Os parâmetros são:

$$n=4,\ k=2,\ n-k=2,\ d=3,\ q=3 \text{ e } q^{n-k}=9.$$

A treliça associada a este código linear e a matriz [M] representando a treliça são mostradas:



TRELLI

[1,0,0]	[1,0,0]	[1,7,4]	[1,3,2]
[0,0,0]	[0,0,8]	[2,8,5]	[0,0,0]
[0,0,0]	[0,6,0]	[3,9,6]	[0,0,0]
[0,0,0]	[0,0,1]	[0,0,0]	[0,0,0]
[0,0,0]	[0,8,0]	[0,0,0]	[0,0,0]
[0,1,0]	[6,0,0]	[0,0,0]	[0,0,0]
[0,0,0]	[0,1,0]	[0,0,0]	[0,0,0]
[0,0,1]	[8,0,0]	[0,0,0]	[0,0,0]
[0,0,0]	[0,0,6]	[0,0,0]	[0,0,0]

A geração da treliça permite a implementação de diver - sos algorítmos para decodificação de códigos de bloco lineares, alguns apresentados nesta tese.

4.2.3 - Curvas de Desempenho

As curvas de desempenho para o algorítmo de maximização da probabilidade a posteriori das palavras código obtidas através de simulação em computador digital estão apresentadas nesta seção. As considerações descritas na página 107 sobre a notação utilizada novamente se aplicam. Estas curvas permitem uma visualização da melhoria obtida com uso de decisão suave. O caso onde decisão suave é utilizada quantizando em Q regiões as amostras recebidas do canal é também considerado.

Figura 4.18 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

Curvas PBIT x SNR

1 - Decisão abrupta NQ = 2

2 - Decisão suave NQ = ∞

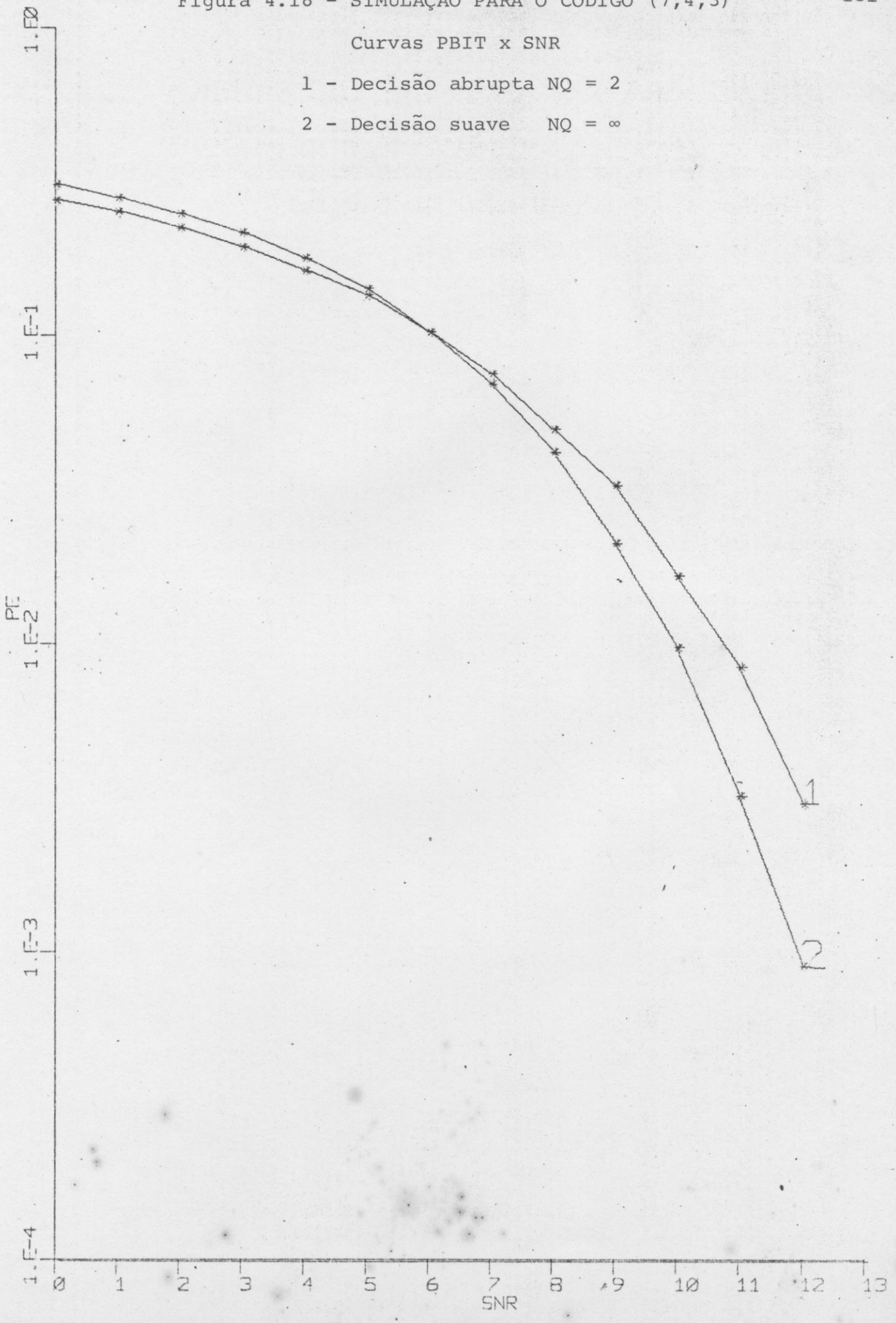


Figura 4.19 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

Curvas PBLOCO x SNR

1 - Decisão abrupta NQ = 2

2 - Decisão suave NQ = ∞

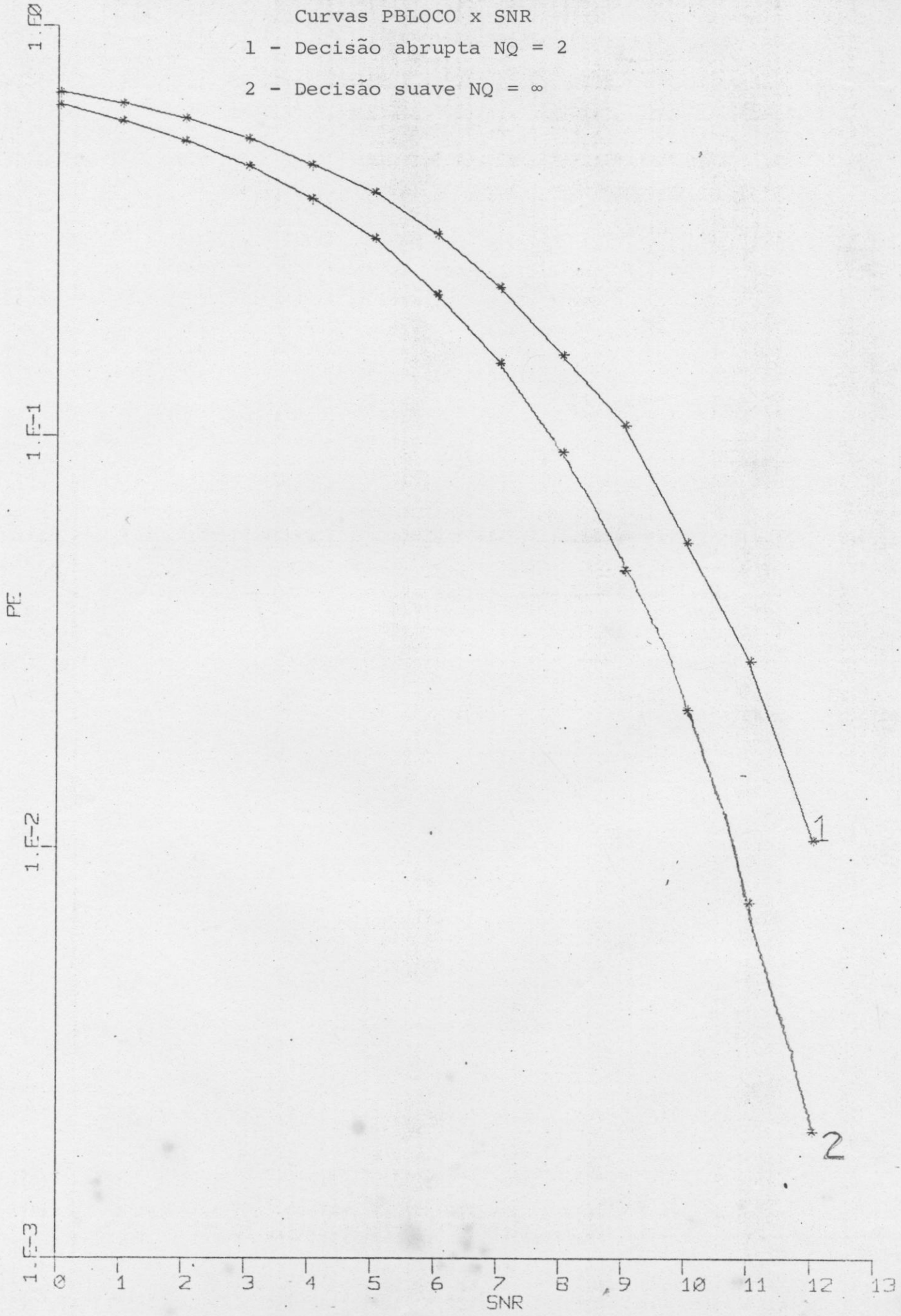


Figura 4.20 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

Curvas PBIT x SNR (Deteção por envoltória)

1 - Decisão abrupta NQ = 2

2 - Decisão suave NQ = ∞

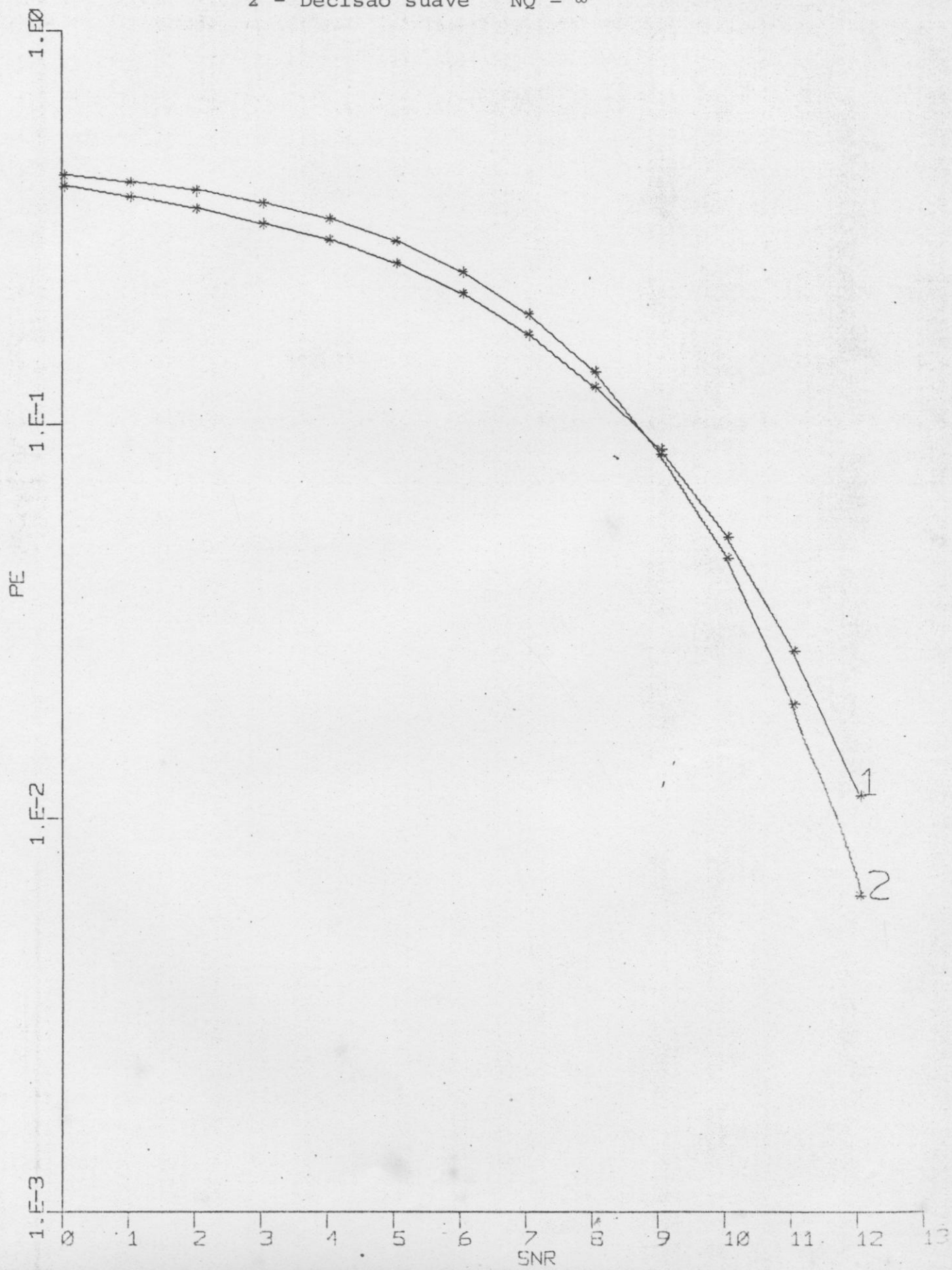


Figura 4.21 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

Curvas PBLOCO x SNR (Deteção por envoltória)

1 - Decisão abrupta NQ = 2

2 - Decisão suave NQ = ∞

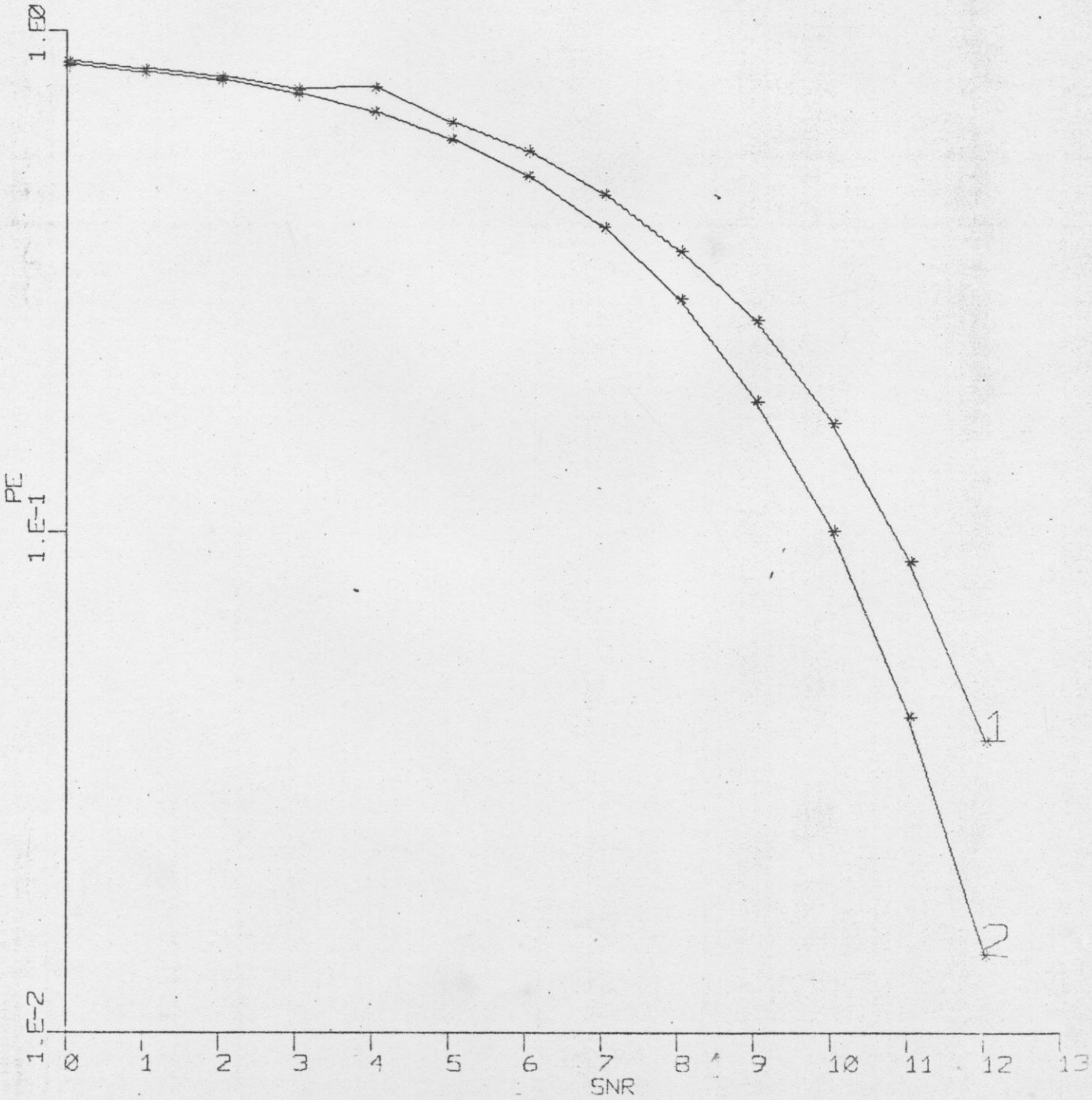


Figura 4.22 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

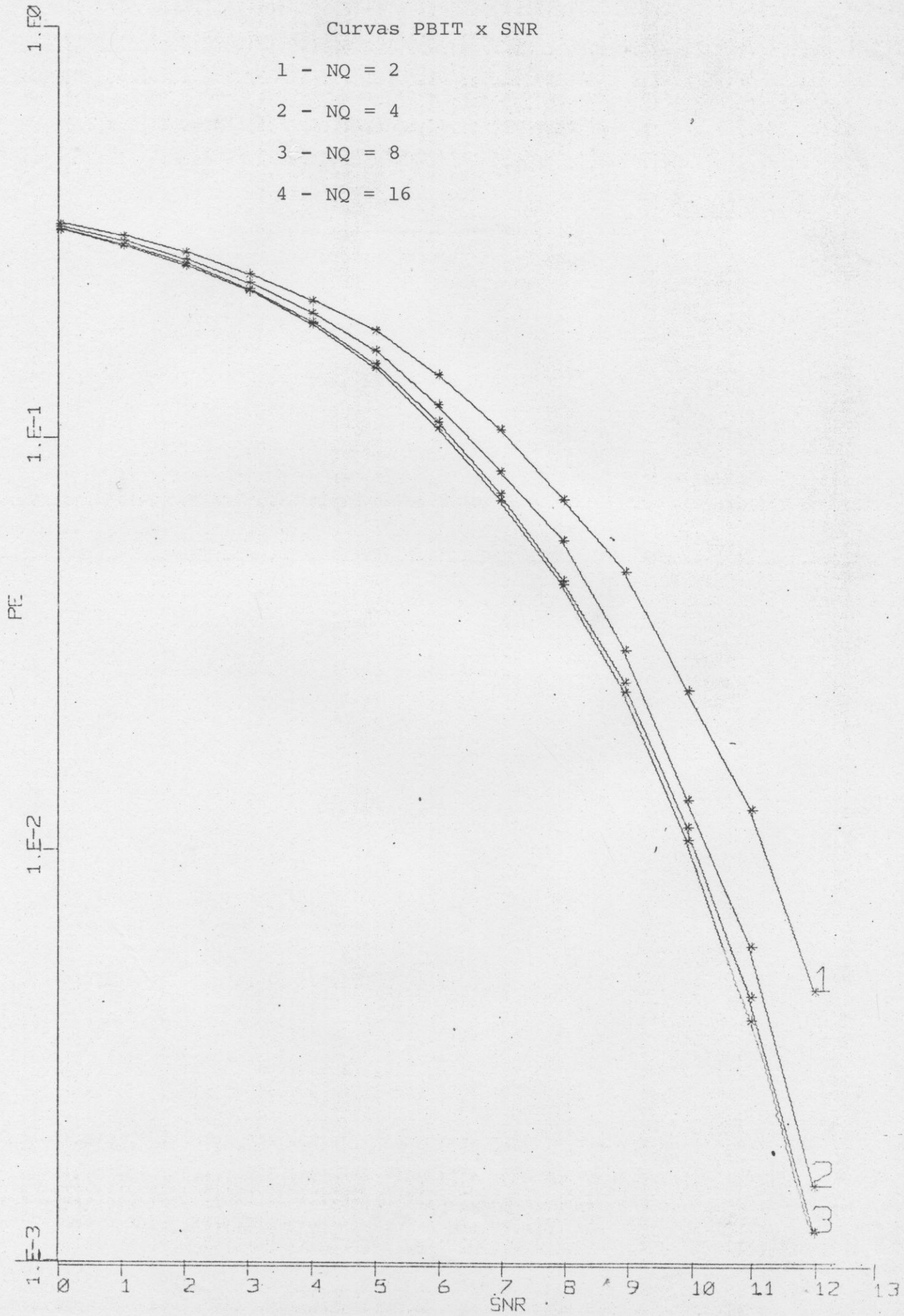
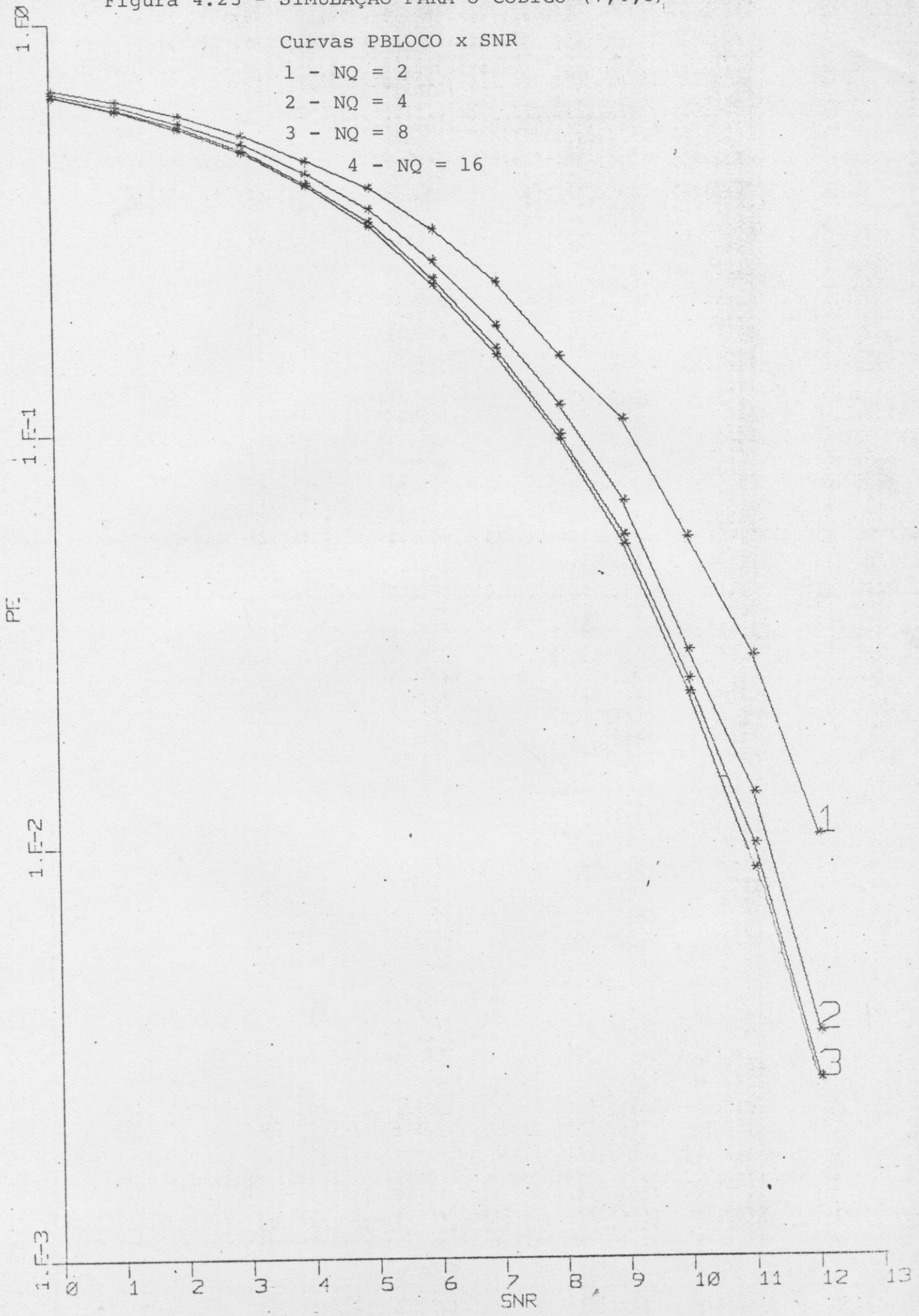


Figura 4.23 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)



CAPÍTULO V

CONCLUSÕES

5.1 - ANÁLISE DOS RESULTADOS

O intuito principal deste trabalho é o de apresentar e analisar procedimentos de decodificação de códigos corretores de erros, em função da melhoria proporcionada no desempenho de sistemas codificados. Como visto, tal melhoria é obtida através do uso da informação probabilística associada as amostras recebidas, e em geral é conseguida as custas de um aumento na complexidade do sistema.

No capítulo II, após a revisão sobre códigos lineares, foram introduzidos procedimentos subótimos que empregam técnicas de decisão suave (sec.2.2), os quais proporcionam uma melhoria considerável no desempenho do sistema, como mostrado nas curvas correspondentes as figuras 2.4, 2.6 e 2.8. Eles são muito atrativos levando-se em consideração que o uso de regiões de quantização é adequado por facilitar a implementação utilizando técnicas digitais. O procedimento proposto por Wolfenson-Rocha é de grande importância, visto que é um dos poucos esquemas de decodificação que assume ser desconhecida a distribuição de probabilidade da fonte. A melhoria provida pelo uso desta técnica encontra-se analisada com detalhes na referência |7|. O desempenho do algorítmo de decodificação por dis -

tância generalizada II foi analisado através de simulação em computador, e comparado com o desempenho do sistema empregando decisão abrupta, resultando nas curvas das figuras 5.1 e 5.2 mostradas a seguir.

Para os procedimentos ótimos descritos no capítulo III, o estabelecimento da regra de decodificação permitiu a determinação do comportamento assintótico para alta relação sinal/ruído. As simulações realizadas para estes algorítmos estão apresentadas nas seções 4.1.2 e 4.2.3 e a obtenção destas curvas são de importância fundamental neste trabalho. A partir de cerca de 12dB, a aproximação utilizando a expressão assintótica fornece excelentes resultados, e é acima deste valor que a simulação se torna praticamente inviável. Deve ser observado que já existiram problemas na convergência da estimação da probabilidade de erro quando utilizando o código (31,26,3), isto porque a regra de decodificação descrita pela equação (3-2) torna-se excessivamente complexa* aumentando de modo sensível o tempo necessário para decodificar um bit. Este fato está refletido em algumas curvas da seção 4.1.2 para os maiores valores da relação sinal/ruído.

Quando o número de níveis de quantização é aumentado, obtém-se informação probabilística mais detalhada para ser utilizada pelo decodificador. Entretanto, um aumento no número de níveis de quantização significa um crescimento na complexidade dos circuitos detetores. Em todos os procedimentos descritos, observa-se que a medida que o valor de NQ aumenta, o ganho incremental na relação sinal/ruído para se obter uma dada probabilidade de erro é cada vez menor, de modo que grande parte da degradação que ocorre quando NQ = 2 pode ser superada sem que a complexidade do sistema se torne proibitiva. A informação adicional obtida por quantizar as amostras recebidas em mais que 16 níveis acrescenta muito pouco àquela obtida para 8 níveis.

* Compare este caso com o descrito no exemplo da página 83.

Figura 5.1 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

Curvas PBIT x SNR

1 - Decisão abrupta NQ = 2

2 - Decodificação por distância generalizada II

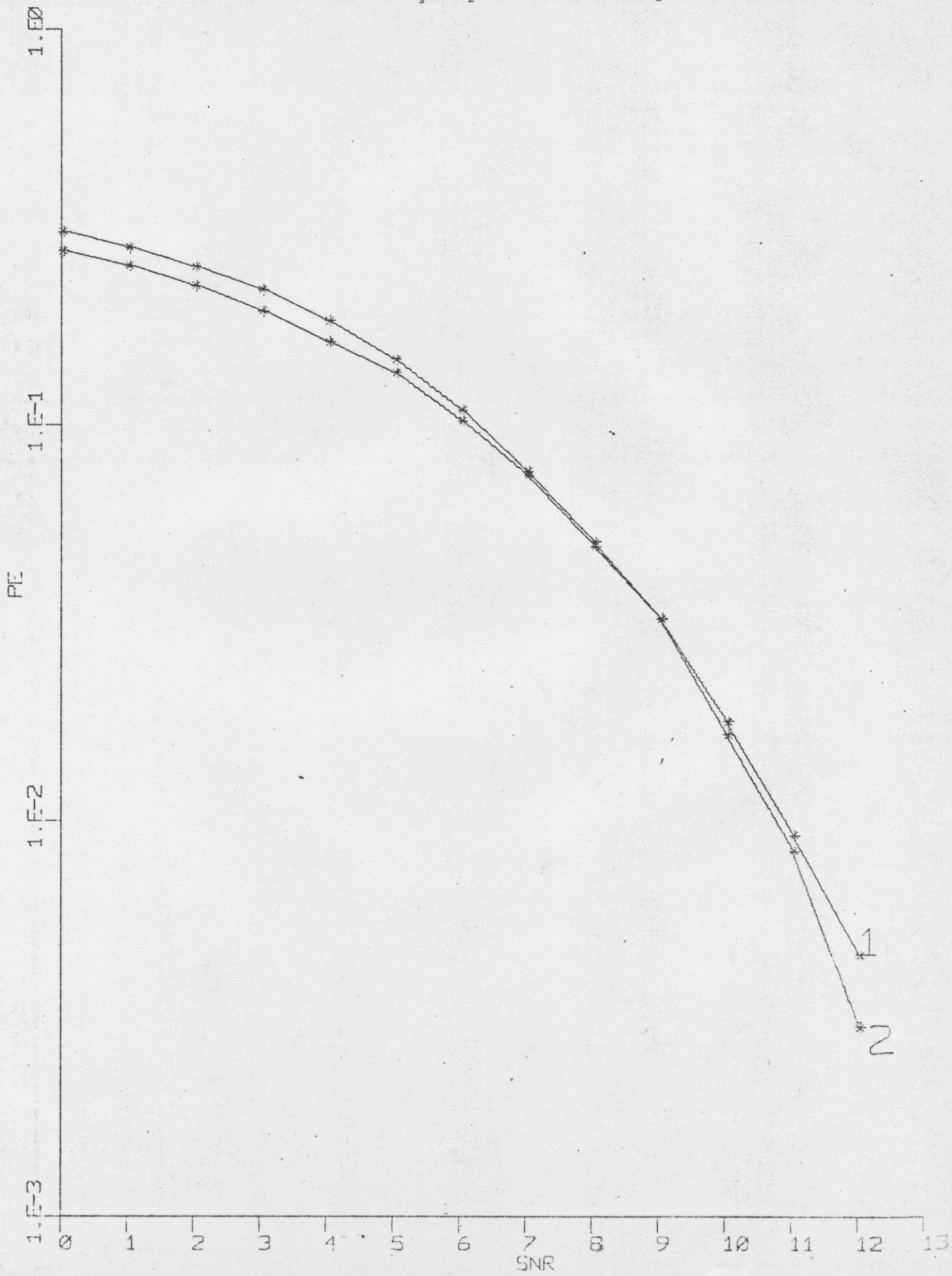
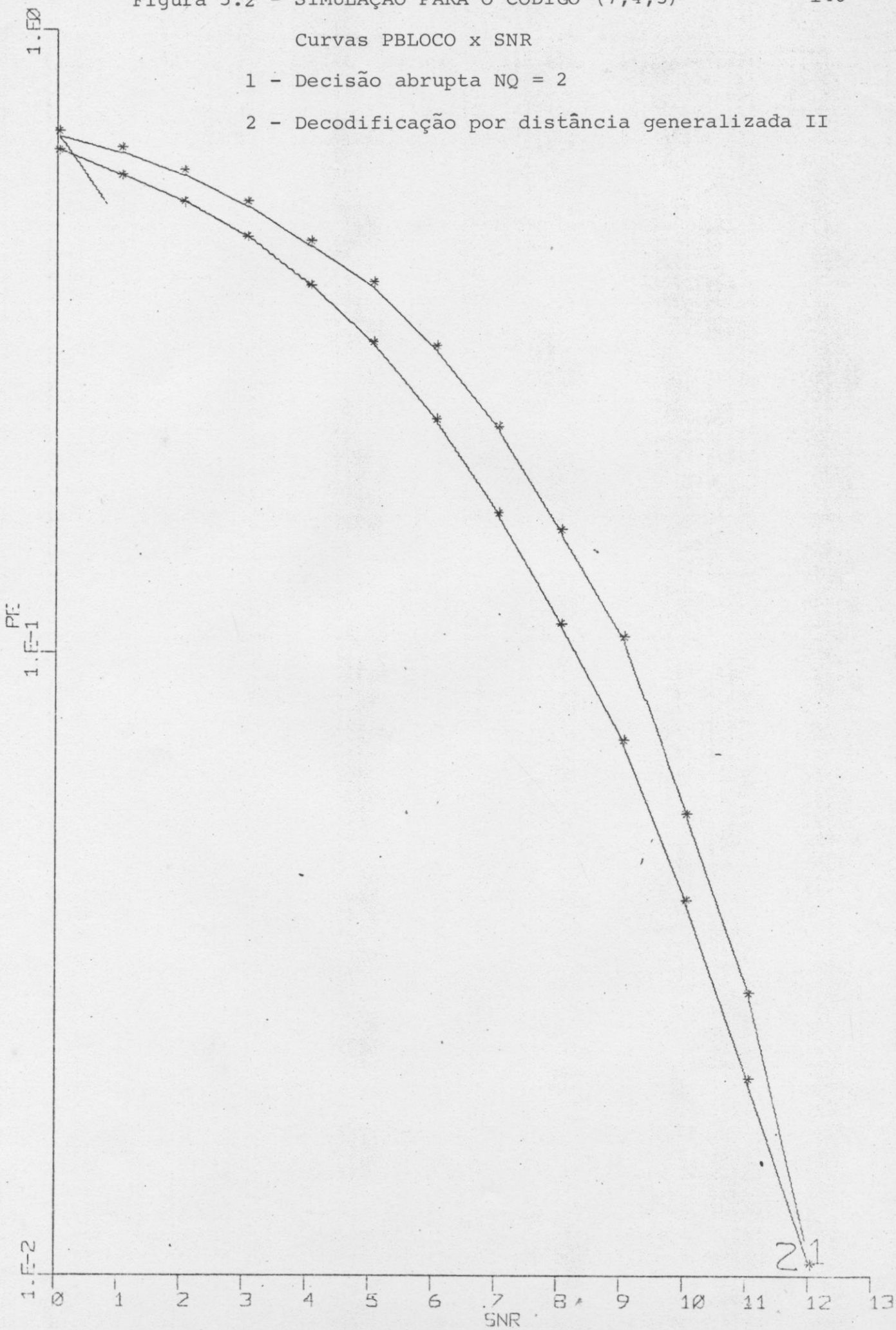


Figura 5.2 - SIMULAÇÃO PARA O CÓDIGO (7,4,3)

Curvas PBLOCO x SNR

1 - Decisão abrupta NQ = 2

2 - Decodificação por distância generalizada II



As curvas citadas na página 105, usadas para verificar a simetria do ruído Gaussiano gerado são apresentadas a seguir.

5.2 - COMENTÁRIOS

Estas técnicas descritas são de particular interesse para sistemas de comunicação via satélite, em comunicação militar e em comunicação espacial de um modo geral. Uma importante e atrativa aplicação está no emprego da decodificação probabilística em sistemas de comunicação utilizados para transmissão de dados em canais de HF, de modo a assegurar uma boa confiabilidade à transmissão.

Neste aspecto, é sugerida uma continuação deste trabalho, propondo-se análise e implementação dos algorítmos estudados, aplicados para canais de HF.

Com relação ao procedimento proposto por J.Wolf descrito no capítulo II, deve ser mencionado que é um procedimento ótimo, equivalente ao algorítmo de maximização da probabilidade a posteriori descrito no capítulo III. Contudo, a formulação matemática no último caso, além de maior rigor, permite a interpretação de "correlação generalizada" e o estabelecimento de cota para a probabilidade de erro por bloco. O mais importante é que permite o uso de decisão suave usando $NQ = 2^Q$ regiões de quantização, o que de acordo com a simulação resulta em considerável melhoria com relação a decisão abrupta e uma simplificação enorme no decodificador. Ademais, comparando-se as expressões utilizadas na decodificação, vê-se que a segunda apresenta uma vantagem computacional.

No caso de códigos de bloco, fica claro que o algorítmo de Hartmann-Rudolph pode ser implementado utilizando a treliça associada ao código dual. A implementação computacional da treli

Figura 5.3 - SIMULAÇÃO PARA NQ = 4

Curvas PBIT x SNR

1 - Palavra toda nula

2 - Palavra toda um

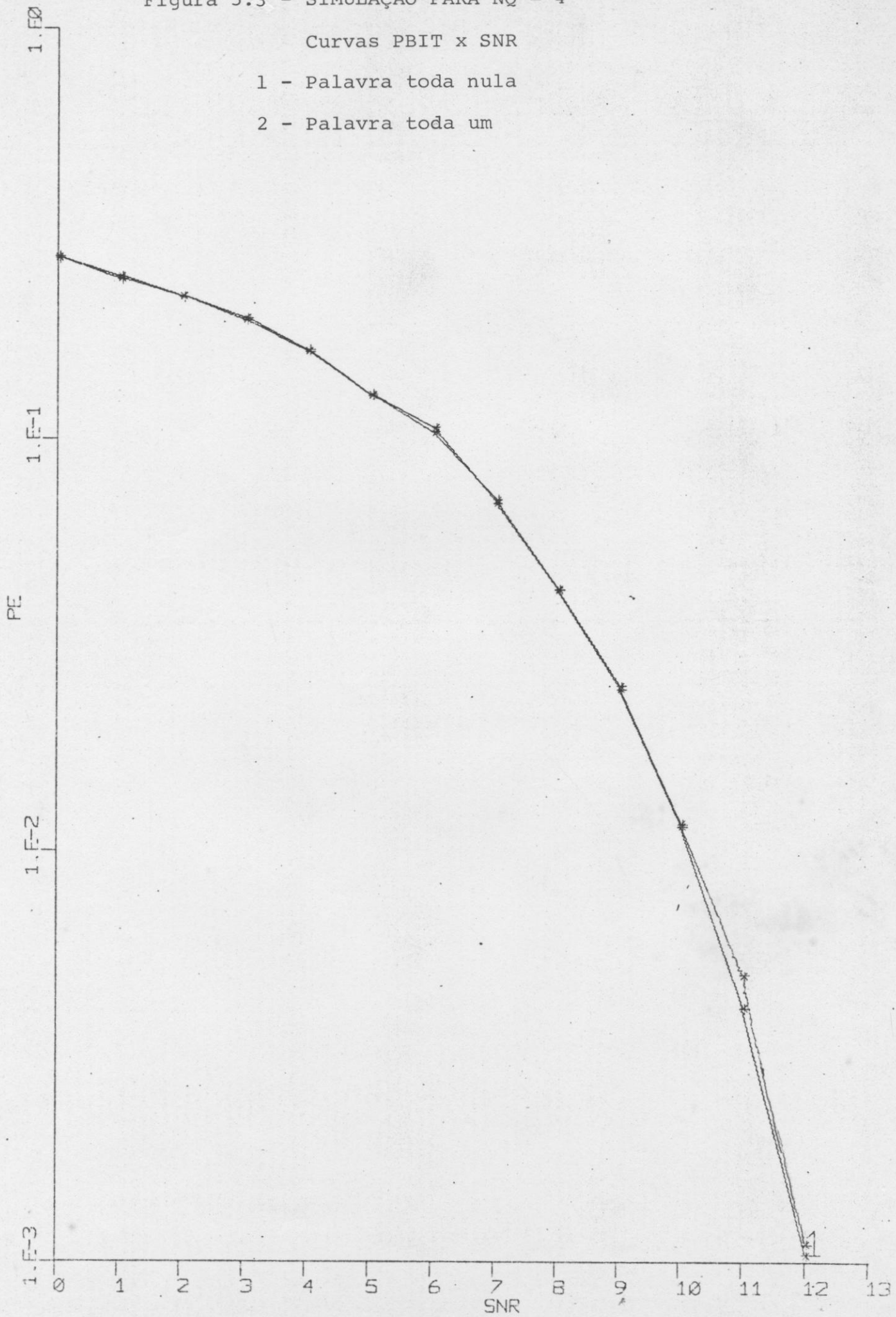
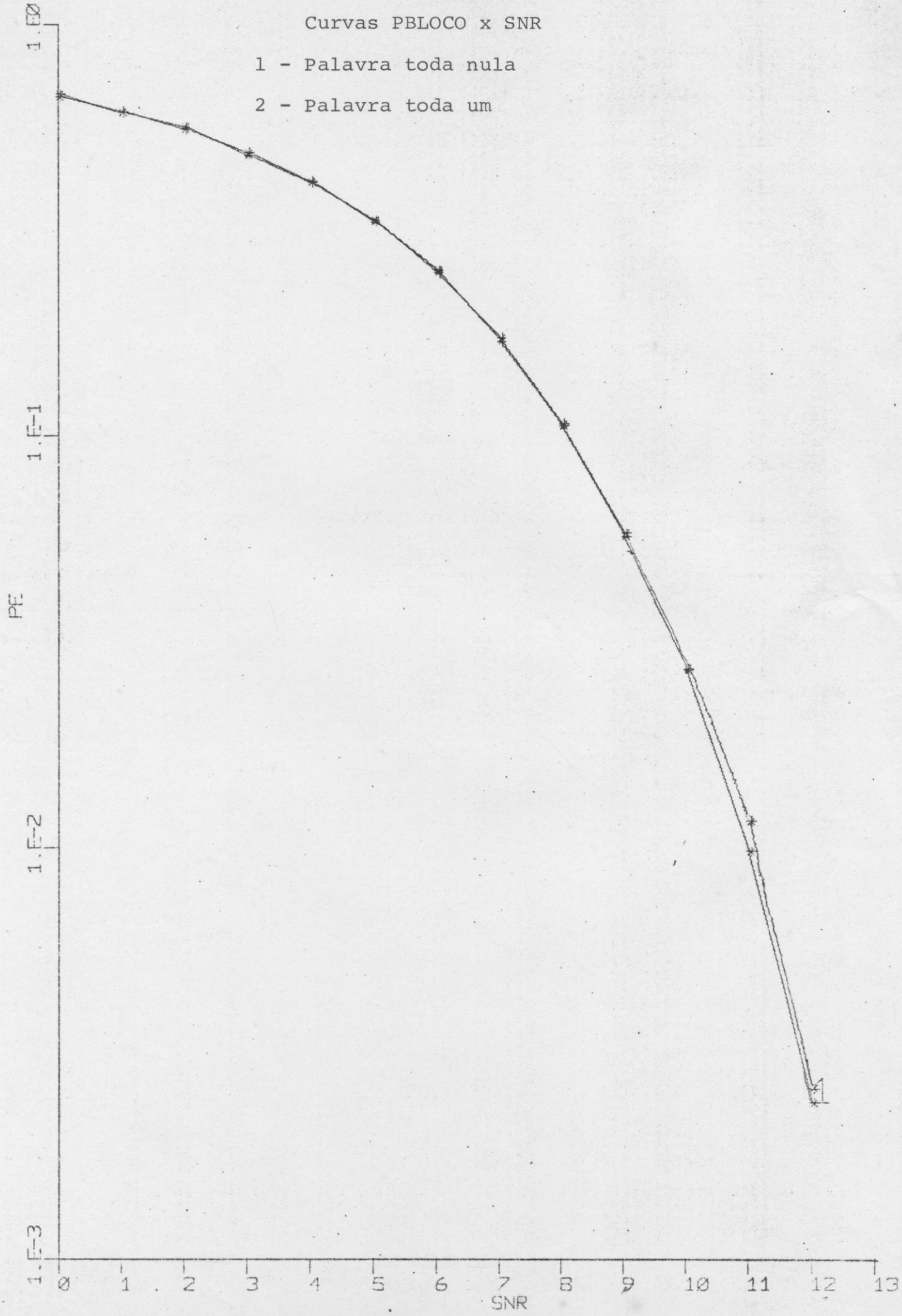


Figura 5.4 - SIMULAÇÃO PARA NQ = 4



ça na forma descrita na seção 4.2.2 e nas subrotinas das listagens do apêndice A, permite a simulação de diversos tipos de decodificadores de códigos de bloco.

Algumas versões dos programas estão apresentadas nas listagens do apêndice A. Entretanto, devido ao tempo de CPU requerido, os programas não podem ser codificados para um caso geral, de modo que devem ser processadas algumas pequenas alterações para cada caso específico.

Ainda como sugestões para pesquisas futuras, dever-se-ia considerar o uso de microprocessadores para implementação dos decodificadores ótimos propostos no capítulo III, permitindo inclusive o emprego de subtreliças. Com relação ao algorítmo de decodificação por distância generalizada II, seria interessante tentar desenvolver uma cota para a probabilidade de erro.

É interessante observar que a decodificação por distância generalizada II resulta em uma palavra onde a médida dos coeficientes de confiabilidades condicionais dos dígitos é máxima. Este fato encontra-se demonstrado no apêndice C.

A medida de confiabilidade associada a uma classe pode ser determinada mesmo no caso onde a partição do espaço de observações resulta em classes de confiabilidade não simétricas, e pode ser calculada no caso multinível. É possível conjecturar que exista uma maneira ótima de combinar os coeficientes de confiabilidade $\rho^{(i)}$ de modo a obter um algorítmo que minimize a probabilidade de erro por símbolo (similar ao descrito no capítulo III) que possa ser aplicável a códigos de baixa eficiência e que seja simples no caso multinível. Uma posterior extensão para decodificação de códigos convolucionais seria bastante útil, pois neste caso os códigos utilizados geralmente são de taxa 1/n.

As técnicas de decodificação apresentadas nesta tese se mostram muito poderosas e práticas em canais que tenham com-

portamento aproximadamente semelhantes a canais sem memória. Com o custo do hardware digital diminuindo rapidamente, o uso destas técnicas crescerá largamente em aplicações práticas em situações de controle de erros.

```
C       UFPE/DES- COMENE
C       DECODIFICACAO DE CODIGOS LINEARES
C       ALGORITMO DE HARTMANN E RUDOLPH
        INTEGER ALL,BIT,BOUND,CHECK,D,DMIN,G,H,W,SM2,T,X
        DIMENSION BOUND(4,4),D(32,31),G(32),H(32),P1(16),PU(16),K(16),KU(3
       -1),X(32)
        COMMON D,NC,K,I,J
        DATA G/32*0/,X/32*0/,YES/'YES'/
        ERF(X)=1.-ERFC(X)
        SQ2=SQRT(2.)
C       ENTRADA DE DADOS : BLOCK CODE
        WRITE(5,11)
11      FORMAT(1H0,'ENTER N,K,D=',/)
        M=20
        OPEN (UNIT=20,FILE='HAND.DAT')
        READ(M,*)NC,KC,DMIN
        N=NC-KC
        IF(NC.GT.32.OR.N.GT.5)STOP
        W=2**N
        KCP1=KC+1
        NCP1=NC+1
        NP1=N+1
        NL1=N-1
        WL1=W-1
        WRITE(5,22)
22      FORMAT(1H0,'CYCLE CODE? YES OR NO.')
        READ(M,33)TESTE
33      FORMAT(A3)
        IF(TESTE.NE.YES)GO TO 1000
C       ROTINA P/ CODIGOS CICLICOS. (POLINOMIO GERADOR).
C       (7,4)    => (3,1,0) = 1,0,1,1
C       (15,11)  => (4,1,0) = 1,0,0,1,1
C       (31,26)  => (5,2,0) = 1,0,0,1,0,1
        WRITE(5,44)
44      FORMAT(1H0,'ENTER G(X)=',/)
        READ(M,*)(G(I),I=1,NP1)
        X(1)=1
        X(NCP1)=1
        DO 200 K=1,NCP1
        IF(X(K).EQ.0)GO TO 100
        H(K)=1
        CALL SMOD2(X,G)
        CALL SHIFT(G)
        GO TO 200
100     CALL SHIFT(G)
200     CONTINUE
        DO 300 I=1,32
300     CHECK=CHECK+X(I)
        IF(CHECK.NE.0)STOP
        DO 500 I=1,WL1
        DO 400 J=1,NC
400     D(I,NCP1-J)=H(J)
        CALL SHIFT(H)
500     CONTINUE
        GO TO 2000
C       ROTINA P/ CODIGOS DE BLOCO NAO-CICLICOS.
1000    WRITE(5,55)
55      FORMAT(1H0,'ENTER (H)',/)
        DO 600 I=1,N
```

NÚCLEO DE PROCESSAMENTO DE DADOS DA UNIVERSIDADE FEDERAL DE PERNAMBUCO

```
66     FORMAT(32(I1,1X))
600    CONTINUE
C      GERANDO PALAVRAS DO CODIGO DUAL.
2000   K=N
       DO 700 I=1,NL1
       BOUND(1,I)=K+1
       IP1=I+1
       DO 700 J=IP1,N
       K=K+1
       CALL SCRAM
700    CONTINUE
       IF(N.LT.3)GO TO 800
       IEND=NL1
       DO 800 L=2,NL1
       IEND=IEND-1
       DO 800 I=1,IEND
       BOUND(L,I)=K+1
       JO=BOUND(L-1,I+1)
       JF=BOUND(L-1,NL1)
       DO 800 J=JO,JF
       K=K+1
       CALL SCRAM
800    CONTINUE
C      ***************************************************************
       T=(DMIN-1)/2
       A=10
       ALL=0
C#     NUMERO DE REGIOES DE QUANTIZACAO = NQ
        DO 50 INN=2,4,1
        NQ=2**INN
       WRITE(3,1)NQ
     1 FORMAT(1H1,49X,'UNIVERSIDADE FEDERAL DE PERNAMBUCO',/,51X,'MESTRAD
      -O EM ENGENHARIA ELETRICA',/,30X,'** DECODIFICACAO DE CODIGOS LINEA
      -RES - ALGORITMO DE HARTMANN & RUDOLPH **',/,5X,'NQ=',I4)
       NQ1=NQ-1
C#     VARIANDO SIGNAL-TO-NOISE RATIO.
       DO 50 SNR=11.,12.,1.
       NY1=0
       Z=10.**(SNR/20.)
C      K=1  (LOW NOISE)
       B=Z*(NQ1)/(NQ*SQ2)
       P1(1)=0.5*ERFC(B)
       B=Z/(NQ*SQ2)
       PO(1)=0.5*(1.+ERF(B))
C      K>  NQ-1  (HIGH NOISE)
       P1(NQ)=PO(1)
       PO(NQ)=P1(1)
C      1<K<NQ-1
       IF(NQ.EQ.2)GO TO 30
       DO 30 K=2,NQ1
       B1=Z*(NQ-K+1)/(NQ*SQ2)
       B2=Z*(NQ-K)/(NQ*SQ2)
       P1(K)=0.5*(ERF(B1)-ERF(B2))
       PO(NQ-K+1)=P1(K)
    30 CONTINUE
C      CALCULO DOS RO'S
       DO 80 I=1,NQ
       R(I)=(1.-P1(I)/PO(I))/(1.+P1(I)/PO(I))
    80 CONTINUE
       SIGMA=A/Z
```

```
      TBE=0.
      TPE=0.
C#
      BIT=(10**4)*NC
      STO=0.
C     SIMULACAO DO ALGORITMO
      DO 40 KDIG=1,BIT,NC
      NY2=0
      DO 10 I=1,NC
C     GERANDO RUIDO GAUSSIANO.
   15 V1=2*RAN(50)-1
      V2=2*RAN(34)-1
      S=V1*V1+V2*V2
      IF (S.GE.1) GO TO 15
      V=V1*SQRT(-2*ALOG(S)/S)
      Y=SIGMA*V
C     POSSIBILIDADES:
      IF(Y.GE.A/2.)NY2=NY2+1
C     IF(Y.LE.-A/2.)NY2=NY2+1
C     INTERVALO ONDE CAIU A AMOSTRA =NY
      NY=(ALL*A+Y)*NQ/A+1
C     PROB Y DIVISIBLE BY A/NQ. IS ZERO
      IF(NY.LT.1)NY=1
      IF(NY.GT.NQ1)NY=NQ
      RO(I)=R(NY)
   10 CONTINUE
      NY1=NY1+NY2
      IF (NY2.LE.T)  GO TO 40
      DO 35 I=1,NC
      SP=0.
      DO 25 J=1,W
      PROD=1
      DO 20 L=1,NC
      DELTIL=1
      IF(I.NE.L)DELTIL=0.
      IF(D(J,L).NE.DELTIL)PROD=PROD*RO(L)
   20 CONTINUE
   25 SP=PROD+SP
C     DECISAO : SP>0. ESTIMAR C^=0, SP<0. ESTIMAR C^=1
C     POSSIBILIDADES:
      IF(SP.LE.0)TBE=TBE+1
C     IF(SP.GT.0)TBE=TBE+1
   35 CONTINUE
       IF(TBE.GT.STO)TPE=TPE+1
      STO=TBE
   40 CONTINUE
C     CALCULANDO PROBABILIDADES DE ERRO.
      PEB=TBE/BIT
      PEP=TPE*NC/BIT
C     SAIDA DOS DADOS
      WRITE(3,2)SNR,TBE,PEB,TPE,PEP,NY1
    2 FORMAT(1H0,15X,'SNR=',E14.8,4X,'TBE=',E14.8,4X,'PEB=',E14.8,4X,'T
     -PE=',E14.8,4X,'PEP=',E14.8,/,'0',37X,'NY1=',I6)
   50 CONTINUE
      STOP
      END
```

```
C       SUBROTINA P/ CALCULO DE ERF - APROX. POLINOMIAL
C       FUNCAO DISTRIBUICAO NORMAL DE PROBABILIDADE
C       ERRO /E(X)/ > 7.5 E-08
C       DEES - DEPARTAMENTO ENGENHARIA ELETRONICA E SISTEMAS
        FUNCTION ERFC(Z)
        X=ABS(SQRT(2.)*Z)
        Q= SQRT(1/(2*3.141593))
        P=  .2316419
        B1= .3193815
        B2=-.3565638
        B3=1.781478
        B4=-1.821256
        B5=1.3302740
C
        T=1/(1+P*X)
        ERFC= Q*EXP((-X**2)/2)
        ERFC= ERFC*T*(B1+T*(B2+T*(B3+T*(B4+T*B5))))
        ERFC=2.*ERFC
        RETURN
        END
        SUBROUTINE SMOD2 (X,G)
        INTEGER X(32),G(32)
        DO 1 I=1,32
      1 X(I)=IABS(X(I)-G(I))
        RETURN
        END
        SUBROUTINE SHIFT(G)
        INTEGER G(32),STO1,STO2
        STO1=G(1)
        DO 1 I=1,31
        STO2=G(I+1)
        G(I+1)=STO1
        STO1=STO2
      1 CONTINUE
        G(1)=STO1
        RETURN
        END
C       SUBROUTINE
        SUBROUTINE SCRAM
        INTEGER D
        DIMENSION D(32,31)
        COMMON D,NC,K,I,J
        DO 1 L=1,NC
      1 D(K,L)=ABS(D(I,L)-D(J,L))
        RETURN
        END
```

```
C       UFPE/DES - COMENF
C       DECODIFICACAO DE CODIGOS LINEARES
C       ALGORITMO DE DECODIFICACAO POR MAXIMA PROBABILIDADE A POSTERIORI
        COMMON GF,N,NPS,J
        INTEGER C(7),GF,GFQ,HAT(7),H(3,7)
        DIMENSION MATRIX(8,7,0:1),MATRIZ(8,7,0:1),PETRIX(8,7,0:1)
        DIMENSION R(7),STAR(7),P1(16),PO(16),ALOGFI(16)
        ERF(X)=1.-ERFC(X)
        SQ2=SQRT(2.)
        NQ=16;NQ1=NQ-1
        GF=1
        GFQ=GF+1
        N=7;K=4
        J=N-K
        NPS=GFQ**J
        NWORDS=10**4
        WRITE(5,33)N,K,GFQ
        WRITE(5,11)
        OPEN(UNIT=20,FILE='H.DAT')
        DO 1 JJ=1,J
        READ(20,*)(H(JJ,NN),NN=1,N)
1       WRITE(5,10)(H(JJ,NN),NN=1,N)
        CALL TRELLI(H,MATRIX,MATRIZ)
        CALL EXPUR(MATRIX,MATRIZ)
        WRITE(5,22)
        DO 2 K1=1,NPS
2       WRITE(5,100)((MATRIX(K1,K2,K3),K3=0,GF),K2=1,N)
        DO 5  SNR=0.,12.,1.
        TBE=0.
        TPE=0.
C
         Z=10.**(SNR/20.)
         B=Z*NQ1/(NQ*SQ2)
         P1(1)=0.5*ERFC(B)
         B=Z/(NQ*SQ2)
         PO(1)=0.5*(1.+ERF(B))
         P1(NQ)=PO(1)
         PO(NQ)=P1(1)
         IF(NQ.LE.2)STOP
         DO 17 IREG=2,NQ1
         B1=Z*(NQ-IREG+1)/(NQ*SQ2)
         B2=Z*(NQ-IREG)/(NQ*SQ2)
         P1(IREG)=0.5*(ERF(B1)-ERF(B2))
         PO(NQ-IREG+1)=P1(IREG)
17       CONTINUE
         DO 231 IREG=1,NQ
231      ALOGFI(IREG)=ALOG(P1(IREG)/PO(IREG))
C
        DO 4 NUM=1,NWORDS
        CALL WORDS(MATRIX,C)
        XSNR=SNR
        CALL NOISE(XSNR,N,R,C,STAR)
        DO 24 KX=1,N
24      STAR(KX)=ALOGFI(IFIX(STAR(KX)))
        CALL CORREL(MATRIX,PETRIX,STAR)
        CALL SURVIV(MATRIX,PETRIX,HAT)
        IERROR=0
        DO 3 KX=1,N
3       IERROR=IERROR+IABS(C(KX)-HAT(KX))
        TBE=TBE+IERROR
```

```
      IF(IERROR.NE.0)TPE=TPE+1
4     CONTINUE
      PEB=TBE/(N*NWORDS)
      PEP=TPE/NWORDS
      WRITE(5,44)SNR,PEB,PEP
5     CONTINUE
C
10    FORMAT(1H ,8X,10(I1,1X))
100   FORMAT(1H0,5X,'.',2X,10('.[',I1,',',I1,']',2X))
11    FORMAT(1H0,8X,'MATRIZ DE VERIFICACAO DE PARIDADE H')
22    FORMAT(1H0,//,6X,'MATRIZ ASSOCIADA A TRELICA ****')
33    FORMAT(1H1,44X,'UNIVERSIDADE FEDERAL DE PERNAMBUCO',/,47X,'MESTRA
     -DO EM ENGENHARIA ELETRICA',/,30X,'** DECODIFICACAO DE CODIGOS LINE
     -ARES - ALGORITMO DE ROCHA JR **',///,9X,'CODIGO DE BLOCO (',I2,','
     -,I2,')',3X,'SOBRE GF(',I1,')',///)
44    FORMAT(1H0,5X,'SNR=',F4.1,5X,'PEB=',E14.8,5X,'PEP=',E14.8)
      STOP
      END
```

```
C       SUBROTINA P/ CALCULO DE ERF - APROX. POLINOMIAL
C       FUNCAO DISTRIBUICAO NORMAL DE PROBABILIDADE
C       ERRO /E(X)/ > 7.5 E-08
C       DEES - DEPARTAMENTO ENGENHARIA ELETRONICA E SISTEMAS
        FUNCTION ERFC(Z)
        X=ABS(SQRT(2.)*Z)
        Q= SQRT(1/(2*3.141593))
        P=  .2316419
        B1= .3193815
        B2=-.3565638
        B3=1.781478
        B4=-1.821256
        B5=1.3302740
C
        T=1/(1+P*X)
        ERFC= Q*EXP((-X**2)/2)
        ERFC= ERFC*T*(B1+T*(B2+T*(B3+T*(B4+T*B5))))
        ERFC=2.*ERFC
        RETURN
        END
C#1TRELLI#2NOCAL#3TRANBD#4TRANDB#5EXPUR#6WORDS#7NOISE#8CORREL#9SURVIV
C
C#1     GERANDO A TRELICA NAO-EXPURGADA
C
        SUBROUTINE TRELLI(H,MATRIX,MATRIZ)
        COMMON GF,N,NPS,J
        INTEGER ALFA,GF,H(J,N),MATRIX(NPS,N,0:GF),MATRIZ(NPS,N,0:GF)
C       GERANDO A TRELLIS 1 (EM MATRIX)
        CALL NOCAL(1,1,1,MATRIX,H)
        DO 1 KX=2,N,1
        MKX=KX
        KY=KX-1
        DO 1 NAT=1,NPS
        MNAT=NAT
        ISUM=0
        DO 13 ALFA=0,GF
13      ISUM=ISUM+MATRIX(MNAT,KY,ALFA)
        IF(ISUM.EQ.0)GO TO 1
        CALL NOCAL(1,MKX,MNAT,MATRIX,H)
1       CONTINUE
C       GERANDO A TRELLIS 2 (EM MATRIZ)
        CALL NOCAL(2,N,1,MATRIZ,H)
        DO 2 KX=N-1,1,-1
        MKX=KX
        KY=KX+1
        DO 2 NAT=1,NPS
        MNAT=NAT
        DO 2 ALFA=0,GF
        IF(MATRIZ(MNAT,KY,ALFA).EQ.0)GO TO 2
        CALL NOCAL(2,MKX,MATRIZ(MNAT,KY,ALFA),MATRIZ,H)
2       CONTINUE
        RETURN
        END
C
C#2     CALCULANDO OS NOS - PREENCHIMENTO DA MATRIZ DA TRELICA
C
        SUBROUTINE NOCAL(NT,KX,AT,MATR,H)
        COMMON GF,N,NPS,J
        INTEGER AT,ALFA,BT(5),CT(5),GF,GFQ,MATR(NPS,N,0:GF),H(J,N)
        ISIG=(-1)**(NT-1)
```

```
      GFQ=GF+1
      CALL TRANDB(GFQ,J,AT,BT)
      DO 2 ALFA=0,GF
      DO 1 J1=1,J
      LL=BT(J1)+ISIG*ALFA*H(J1,KX)
      CT(J1)=MOD(LL,GFQ)
      IF(CT(J1).LT.0)CT(J1)=CT(J1)+GFQ
1     CONTINUE
      CALL TRANBD(GFQ,J,NO,CT)
      IF(NT.EQ.1)MATR(NO,KX,ALFA)=AT
      IF(NT.EQ.2)MATR(AT,KX,ALFA)=NO
2     CONTINUE
      RETURN
      END
C
C#3   TRANSFORMACAO BD
C
      SUBROUTINE TRANBD(GFQ,J,DD,BB)
      INTEGER DD,BB(5),GFQ
      DD=0
      DO 1 M=1,J
      DD=DD+BB(J+1-M)*GFQ**(M-1)
1     CONTINUE
      DD=DD+1
      RETURN
      END
C
C#4   TRANSFORMACAO DB
C
      SUBROUTINE TRANDB(GFQ,J,DD,BB)
      INTEGER DD,BB(5),GFQ,Q
      Q=DD-1
      DO 1 M=1,J
      BB(J+1-M)=MOD(Q,GFQ)
      Q=Q/GFQ
1     CONTINUE
      RETURN
      END
C
C#5   GERANDO A TRELICA EXPURGADA
C
      SUBROUTINE EXPUR(MATRIX,MATRIZ)
      COMMON GF,N,NPS,J
      INTEGER ALFA,GF,MATRIX(NPS,N,0:GF),MATRIZ(NPS,N,0:GF)
      DO 1 NAT=1,NPS
      DO 1 KX=1,N
      DO 1 ALFA=0,GF
      IF(MATRIX(NAT,KX,ALFA).NE.MATRIZ(NAT,KX,ALFA))
     : MATRIX(NAT,KX,ALFA)=0
      MATRIZ(NAT,KX,ALFA)=0
1     CONTINUE
      RETURN
      END
C
C#6   GERANDO PALAVRAS-CODIGO ALEATORIAMENTE
C
      SUBROUTINE WORDS(MATRIX,C)
      COMMON GF,N,NPS,J
      INTEGER ALFA,C(N),GF,MATRIX(NPS,N,0:GF)
      GFQ=FLOAT(GF+1)
```

```
      NO=1
      DO 1 KX=N,1,-1
10    ALFA=RAN(5)*GFQ
      IF(ALFA.EQ.GFQ)STOP
      IF(MATRIX(NO,KX,ALFA).EQ.0)GO TO 10
      C(KX)=ALFA
      NO=MATRIX(NO,KX,ALFA)
1     CONTINUE
      RETURN
      END
C
C#7   SIMULANDO CANAL RUIDOSO - ENCONTRANDO VETOR RECEBIDO
C
      SUBROUTINE NOISE(SNR,N,R,C,STAR)
      INTEGER C(N)
      DIMENSION R(N),STAR(N)
      NQ=16;NQ1=NQ-1
      A=10.
      Z=10.**(SNR/20.)
      SIGMA=A/Z
      VAR=SIGMA**2.
      DO 2 I=1,N
1     X=2.*RAN(50)-1.
      Y=2.*RAN(34)-1.
      S=X*X+Y*Y
      IF(S.GE.1.)GO TO 1
      V=SIGMA*SQRT(-2.*ALOG(S)/S)
      X=X*V
      Y=Y*V
      R(I)=C(I)*A+X
      NY=R(I)*NQ/A+1
      IF(NY.LT.1)NY=1
      IF(NY.GT.NQ1)NY=NQ
      STAR(I)=NY
2     CONTINUE
      RETURN
      END
C
C#8   DETERMINANDO A CORRELACAO GENERALIZADA
C
      SUBROUTINE CORREL(MATRIX,PETRIX,VECTOR)
      COMMON GF,N,NPS,J
      INTEGER ALFA,GF
      DIMENSION MATRIX(NPS,N,0:GF),PETRIX(NPS,N,0:GF),VECTOR(N)
      EPSLON=1.E-38
      JJ=0
      DO 5 J2=1,N
      DO 5 J1=1,NPS
      DO 2 J3=0,GF
      IF(MATRIX(J1,J2,J3).EQ.0)GO TO 2
      IF(J2.NE.1)GO TO 1
      PETRIX(J1,J2,J3)=J3*VECTOR(J2)+EPSLON
      GO TO 2
1     DO 10 ALFA=0,GF
      IF(PETRIX(MATRIX(J1,J2,J3),J2-1,ALFA).NE.0.0)JJ=ALFA
10    CONTINUE
      PETRIX(J1,J2,J3)=PETRIX(MATRIX(J1,J2,J3),J2-1,JJ)
     :+J3*VECTOR(J2)+EPSLON
2     CONTINUE
      PMAX=PETRIX(J1,J2,0)
```

```
      DO 13 J3=0,GF
      PMAX=AMAX1(PMAX,PETRIX(J1,J2,J3))
13    CONTINUE
      IF(PMAX.EQ.0.0)GO TO 5
      DO 5 J3=0,GF
      IF(PMAX.EQ.PETRIX(J1,J2,J3))GO TO 5
      PETRIX(J1,J2,J3)=0.0
5     CONTINUE
      RETURN
      END
C#9   CALCULANDO A PALAVRA DECODIFICADA(SEQUENCIA SOBREVIVENTE)
C
      SUBROUTINE SURVIV(MATRIX,PETRIX,HAT)
      COMMON GF,N,NPS,J
      INTEGER ALFA,GF,HAT(N),MATRIX(NPS,N,0:GF)
      DIMENSION PETRIX(NPS,N,0:GF)
      IND=1
      DO 2 KX=N,1,-1
      DO 1 ALFA=0,GF
      IF(PETRIX(IND,KX,ALFA).NE.0.0)HAT(KX)=ALFA
1     CONTINUE
      IND=MATRIX(IND,KX,HAT(KX))
2     CONTINUE
      RETURN
      END
```

APÊNDICE B

Seja um código de bloco linear binário $C(n,k,d)$ com $m = 2^k$ palavras códigos, denotadas $\underline{c}_i$ $i = 0,1,2,\ldots,m-1$. As palavras são transmitidas pela fonte equiprovavelmente através de um canal perturbado com ruído aditivo branco gaussiano com densidade espectral de potência unilateral $N_o/2$. Então

$$P_{Bloco} = P\left[\varepsilon \,|\, \underline{c}_o\right] = 1 - P\left[C \,|\, \underline{c}_o\right] \tag{B-1}$$

Como é considerado que a énupla transmitida foi $\underline{c}_o$, a palavra toda zero, a decodificação por máximo de verossimilhança será correta se e somente se

$$\sum_{\ell=0}^{n-1} c_{i\ell} \ \ell n \ \phi_\ell < 0 \text{ para cada } i=1,2,3,\ldots,m-1 \tag{B-2}$$

Se os sinais são transmitidos através de pulsos liga-desliga com amplitude $+\sqrt{E}$ Volts ou 0 Volts, então

$$\sum_{\ell=0}^{n-1} c_{i\ell} \ (r_\ell - \sqrt{E}/2) < 0 \quad \forall \ \underline{c}_i \ \varepsilon \ C, \ \underline{c}_i \neq \underline{c}_o. \tag{B-3}$$

Definindo eventos A_i, para $i = 1,2,\ldots,m-1$,

$$A_i \triangleq \left\{ \sum_{\ell=0}^{n-1} c_{i\ell} \ n_\ell < \sqrt{E}/2 \sum_{\ell=0}^{n-1} c_{i\ell} \right\} \tag{B-4}$$

tem-se que (B-1) pode ser reescrita como (B-5) usando (B-2),

$$P_{Bloco} = 1 - P\left[\bigcap_{i=1}^{m-1} A_i\right] \tag{B-5}$$

Por outro lado, $W(\underline{c}_i) = \sum_{\ell=0}^{n-1} c_{i\ell}$ é o peso da palavra $\underline{c}_i$ e $\gamma = R\,\gamma_b = E/N_o$ é a relação sinal/ruído do canal.

Sejam variáveis aleatórias $\{N_i\}_j^{m-1}$, com distribuição normal

$$N_i \triangleq \frac{\sum_{\ell=0}^{n-1} c_{i\ell}\, n_\ell}{\left(\sum_{\ell=0}^{n-1} c_{i\ell} N_o/2\right)^{1/2}} \Rightarrow N_i \sim \mathcal{N}(0,1). \qquad \text{(B-6)}$$

os eventos A_i podem ser agora expressos sob a forma

$$A_i = \{N_i < \sqrt{R\,\gamma_b\, W(\underline{c}_i)/2}\} \quad i = 1,2,\ldots,m-1 \qquad \text{(B-7)}$$

As variáveis aleatórias $\{N_i\}_1^{m-1}$ são conjuntamente gaussianas, com matriz de correlação $\lambda=(\lambda_{ij})$, onde $\lambda_{ij}=E\{N_i.N_j\}$, portanto a correlação λ_{ij} será

$$\lambda_{ij} = \frac{E\{\sum_{\ell=0}^{n-1} c_{i\ell} n_\ell \sum_{k=0}^{n-1} c_{jk}\, n_k\}}{\sqrt{W(\underline{c}_i)}.\sqrt{W(\underline{c}_j)}.N_o/2} \qquad \text{(B-8)}$$

Usando o fato que $\{n_\ell\}_1^{n-1}$ são variáveis aleatórias com variância $\overline{n_\ell^2} = N_o/2$, tem-se que

$$\lambda_{ij} = \frac{\sum_{\ell=0}^{n-1} c_{i\ell} c_{j\ell}\, E\{n_\ell^2\}}{\sqrt{W(\underline{c}_i)}.\sqrt{W(\underline{c}_j)}.N_o/2} = \frac{\underline{c}_i.\underline{c}_j}{\sqrt{W(\underline{c}_i)}.\sqrt{W(\underline{c}_j)}} \qquad \text{(B-9)}$$

Notando que $||c_i|| = (\underline{c}_i.\underline{c}_i)^{1/2} = (W(\underline{c}_i))^{1/2}$, segue-se que

$$\lambda_{ij} = \frac{\underline{c}_i \cdot \underline{c}_j}{||\underline{c}_i||\ ||c_j||} = \cos \widehat{c_i, c_j} \qquad \text{(B-10)}$$

Deve ser observado que $0 \leq \lambda_{ij} \leq 1$, e $\lambda_{ij} = 0$ $\Leftrightarrow$ $\underline{c}_i \perp \underline{c}_j$, ademais $\lambda_{ii} = 1$.

Desta maneira (B-5) pode ser reescrita como sendo

$$P_{Bloco} = 1 - F_{N_1 N_2 \ldots N_{m-1}}(\sqrt{R\gamma_b W(\underline{c}_1)/2}, \ldots, \sqrt{R\gamma_b W(\underline{c}_{m-1})/2}) \qquad \text{(B-11)}$$

$F_{\underline{N}}$ é a função distribuição correspondente a m-1 variáveis aleatórias conjuntamente Gaussianas com matriz de covariância $\lambda = (\lambda_{ij})$.

No cálculo da probabilidade de erro por bloco para o código (15,11,3), por exemplo, é necessário a avaliação da função distribuição conjunta de 2047 variáveis aleatórias dependentes.

APÊNDICE C

LEMA: O algorítmo de decodificação por distância mínima generalizada escolhe a palavra código que maximiza a confiabilidade média dos símbolos.

PROVA: A partir das amostras recebidas do canal, dois vetores são determinados,

$$\underline{r} = (r_1,\ldots,r_n) \qquad R_S^{(\ )} = (R_S^{(1)},\ldots,R_S^{(n)})$$

palavra recebida vetor de confiabilidades

Aqui $R_S^{(i)}$ é o coeficiente de confiança condicional de Kiefer associado à decisão r_i.

Se uma palavra código $\underline{f}$ é assumida com tendo sido transmitida, a confiabilidade associada aos seus dígitos é expressa por

$$R_S^{(i)} \quad \text{se} \quad d_H(r_i,f_i) = 0$$

$$1-R_S^{(i)} \quad \text{se} \quad d_H(r_i,f_i) = 1$$

Deste modo, a confiabilidade média dos símbolos da palavra, admitindo que a palavra transmitida foi $\underline{f}$, é

$$\overline{R_S} = \frac{1}{n} \sum_{i=1}^{n} d_H(r_i,f_i)\{1-R_S^{(i)}\} + \{1-d_H(r_i,f_i)\}R_S^{(i)}$$

Um algorítmo de decodificação que maximize a confiabilidade média das decisões deve escolher uma palavra $\underline{f}$ de acordo com

$$\max_{\underline{f}\varepsilon C} \sum_{i=1}^{n} d_H(r_i,f_i)\{1-R_S^{(i)}\} + \{1-d_H(r_i,f_i)\}R_S^{(i)}$$

$$\max_{\underline{f}\varepsilon C} \sum_{i=1}^{n} R_S^{(i)} - \sum_{i=1}^{n} d_H(r_i,f_i)\{2R_S^{(i)}-1\}$$

Lembrando que $\rho^{(i)} = 2R_S^{(i)} - 1$, obtém-se

$$\underset{\underline{f}\varepsilon C}{\text{Max}} - \sum_{i=1}^{n} d_H(r_i, f_i)\rho^{(i)}$$

o que equivale a $\underset{\underline{f}\varepsilon C}{\text{Min}} \sum_{i=1}^{n} d_H(r_i, f_i)\rho^{(i)}$

A seguir são apresentados dois exemplos ilustrativos. A fonte, o canal e código são admitidos os mesmos do exemplo da página 73.

<u>Exemplo 1</u> - Admitindo que as amostras foram(-0.7,-0.2,-0.3,+0.7,+1.1) tem-se que

$\underline{r} = (0,0,0,1,1)$ e $R_S^{(i)} = (.80,.60,.65,.80,.90)$.

Daí pode ser encontrado $\underline{\rho}^{(\ )} = (.6,.2,.3,.6,.8)$.

A palavra recebida $\underline{r}$ não é uma palavra código. Com relação a duas palavras código $\underline{f}_1 = (1,0,0,1,1)$ e $\underline{f}_2 = (0,1,1,1,1)$, como exemplo, qual delas seria preferível? Aplicando-se o algorítmo, a palavra $\underline{f}_2$ seria escolhida:

$$A(\underline{r},\underline{f}_1) = \rho^{(1)} = .6$$

$$A(\underline{r},\underline{f}_2) = \rho^{(2)} + \rho^{(3)} = .2+.3 = .5$$

Com relação aos coeficientes de confiança condicional

$\underline{f}_1 \Rightarrow R_1^{(\ )} = (.20,.60,.65,.80,.90)$ Média $R_1^{(i)} = 3.15/5 = .63$

$\underline{f}_2 \Rightarrow R_2^{(\ )} = (.80,.40,.35,.80,.90)$ Média $R_2^{(i)} = 3.25/5 = .65$

<u>Exemplo 2</u> - Considerando um segundo caso onde as amostras recebidas' são (-0.7,-0.3,-0.3,+0.7,+1.1), tem-se

$\underline{r} = (0,0,0,1,1)$ e $\underline{R}_S^{(\ )} = (.80,.65,.65,.80,.90)$,

portanto $\underline{\rho}^{(\ )} = (.6,.3,.3,.6,.8)$

$$A(\underline{r},\underline{f}_1) = \rho^{(1)} = .6$$

não há preferência.

$$A(\underline{r},\underline{f}_2) = \rho^{(2)} + \rho^{(3)} = .3 + .3 = .6$$

Com relação aos coeficientes de Kiefer,

$\underline{f}_1 \Rightarrow R_1^{(\ )} = (.20,.65,.65,.80,.90)$ Média $R_1^{(i)} = 3.2/5 = .64$

$\underline{f}_2 \Rightarrow R_2^{(\ )} = (.80,.35,.35,.80,.90)$ Média $R_2^{(i)} = 3.2/5 = .64$

É realmente indiferente para o valor da confiabilidade média dos símbolos se o algorítmo escolhe $\underline{f}_1$ ou $\underline{f}_2$ como sendo a palavra decodificada.

BIBLIOGRAFIA E REFERÊNCIAS

|1| - ALMEIDA,M.A.M., "Decodificação de Códigos Convolucionais", , Tese de Mestrado, Universidade Federal de Pernambuco. (Em preparação).

|2| - BARTLE,R.G., "The Elements of Real Analysis", John Wiley & Sons, Inc., 1964.

|3| - BERLEKAMP,E.R., "Algebraic Coding Theory", McGraw-Hill Book Company, New York, 1968.

|4| - BLOOM,F.J., CHANG,S.S.L., HARRIS,B., HAMPTSCHEIN,A. and MORGAM, K.C., "Improvement of Binary Transmission by Null Zone Reception", IRE procs., vol.45, 1957.

|5| - CAMPELLO DE SOUZA,R.M., "Decodificação Probabilística de Códigos Lineares", Tese de Mestrado, Universidade Federal de Pernambuco, 1979.

|6| - CARLSON,A.B., "An Introduction to Signal and Noise in Electrical Communication", McGraw-Hill Kogakusha, ltda., 2nd edition , 1975.

|7| - CASANOVA,F.A., "Utilização de uma Medida de Conclusividade no Processo de Decodificação: Coeficientes de Confiança. Estudo Comparativo", Trabalho de Graduação, Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, 1982.

|8| - CHERNOFF,H., "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations", Ann.Math.Stat. , Vol.23, pp.493-507, dec.1952.

|9| - DAVENPORT JR.,W.B., "Random Processes - An Introduction for Applied Scientists and Engineers", McGraw-Hill Book Company , 1970.

|10| - DAVENPORT JR.,W.B., ROOT,W.L., "An Introduction to the Theory of Random Singnals and Noise", McGraw-Hill Book Company,1958.

|11| - FARRELL,P.G., "Lectures Notes on Communication Theory", COMENE, Universidade Federal de Pernambuco, junho 1977.

|12| - FARRELL,P.G., MUNDAY,E., and KALLIGEROS,K., "Digital Communication Using Soft Decision Detection Techniques", AGARD Symp. on Dig. Comms in Avionics, Munich, pp.141/9 , june 1978.

|13| - FERGUSON,T.S., "Mathematical Statistics", Academic Press. , 1967.

|14| - FORNEY JR., G.D., "Generalized Minimum Distance Decoding" , IEEE Trans. Inform. Theory, vol. IT-12, pp. 125-131, April ' 1966.

|15| - GALLAGER,R.G., "Information Theory and Reliable Communication", John Wiley & Sons., Inc., 1968.

|16| - HAMMING,R.W., "Error Detecting and Error Correcting Codes" , Bell Systemes Tech.J., 29, pp. 147-160, April 1950.

|17| - HARRISON,C.N., "Application of Soft Decision Techniques to Block Codes", proc.IERE Conference on Digital Processing of Signals in Communication, Loughborough, England, nº 37, 1977.

|18| - HARTMANN,C.R.P., RUDOLPH,L.D., "An Optimum Symbol-by-Symbol Decoding Rule for Linear Codes", IEEE Trans.Inform.Theory , Vol.IT-22, nº 5, pp. 514-517, sep. 1976.

|19| - HARTMANN,C.R.P., RUDOLPH,L.D., and MEHROTRA,K.G.,"Asymptotic Performance of Optimum Bit-by-Bit Decoding for the White Gaussian Channel", IEEE Trans. Inform. Theory, vol. IT-22 , nº 5, pp. 520-522, july 1977.

|20| - HERSTEIN,I.N., "Topics in Algebra", Blaisdell Publishing Company, Massachussets, 1964.

|21| - KIEFER,J., "Conditional Confidence Statements and Confidence Estimators", J.ASA, 72, pp. 789-827, 1977.

|22| - KNUTH,D.E., "The Art of Computer Programming Seminumerical Algorithms - Vol.II", Addison-Wesley series in computer science and information processing, mass. 1973.

|23| - LATHI,B.P., "Random Signals and Communication Theory", International text book company, Pennsylvania, 1968.

|24| - LUCKY,R.W., SALZ,J., WELDON JR., E.J., "Principles of Data Communication", Bell telephone laboratories, Inc., McGraw - Hill Book Company, 1968.

|25| - MATIS,K.R., MODESTINO.J.W., "Reduced-Search Soft-Decision Trellis Decoding of Linear Block Codes", IEEE Trans.Inform. Theory, vol.IT-28, nº 2, March 1982.

|26| - PETERSON,W.W., WELDON JR.,E.J., "Error Correcting Codes", MIT Press., 2nd Edition, Massachussets, 1972.

|27| - ROCHA JR.V.C., "Decoding Using Soft-Decision", DES publica - ção nº 4, Universidade Federal de Pernambuco, dez.1979.

|28| - ROCHA JR.V.C., "Versatile Error Control Coding Systems", Ph.D.Thesis, University of Kent at Canterbury, England, 1976.

|29| - ROCHA JR.,V.C., "An Optimum Soft-Decision Receiver for Block Codes", Comunicação privada, 1982.

|30| - ROCHA JR.V.C., "Códigos Corretores de Erros, Minicurso apresentado no V Congresso nacional da SBMAC, João Pessoa, agosto 1982.

|31| - ROCHA JR.V.C., OLIVEIRA,H.M., Receptor Ótimo para Códigos de Bloco usando Decisão Suave", Comunicação apresentada no V Congresso da SBMAC, João Pessoa, agosto 1982.

|32| - RUDIN,W., "Principles of Mathematical Analysis", McGraw-Hill book company, 1964.

|33| - SCHWARTZ,M., "Information Transmission, Modulation and Noise", McGraw-Hill kogakusha, ltd., 2nd edition, 1970.

|34| - SCHWARTZ,M., BENNETT.W.R., STEIN,S., "Communications Systems and Techniques", McGraw-Hill book company, 1966.

|35| - SHANNON,C., "A Mathematical Theory of Communication", Bell systems tech.j., vol.27, (pt.I), pp.374-427, (pt.II), pp.623-656, 1948.

|36| - SHU LIN, "An Introduction to Error-Correcting Codes", Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1970.

|37| - VITERBI,A.J., OMURA,J.V., "Principles of Digital Communications and Coding", McGraw-Hill book company, 1979.

|38| - WOLF,J.K., "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis", IEEE Trans.Inform.Theory, vol. IT-24, nº 1, pp. 76-80, jan. 1978.

|39| - WOLFENSON,M., ROCHA JR.V.C., "Soft Decision Decoding with Unknow Source Distribution", Electronics Letters, vol.16, nº 25/26, dec. 1980.

|40| - WOZENCRAFT,J.M., JACOBS,I.M., Principles of Communication Engineering", John Wiley & Sons, Inc. New York, 1967.